

UTILITY PATENT APPLICATION TRANSMITTAL

Submit an original and a duplicate for fee processing (Only for new nonprovisional applications under 37 CFR §1.53(b))

ADDRESS TO:

Attorney Docket No.

204843

Assistant Commissioner for Patents

First Named Inventor

GILIBERTO

Box Patent Application Washington, D.C. 20231

Express Mail No.

EL304646376US

	<u> </u>
APPLICATION ELEMENTS	ACCOMPANYING APPLICATION PARTS
 Utility Transmittal Form Specification (including claims and abstract) [Total Pages 19] 	Assignment Papers (cover sheet and document(s)) Power of Attorney
3. Drawings [Total Sheets 4] 4. Combined Declaration and Power of Attorney [Total Pages] a. Newly executed b. Copy from prior application	10. English Translation Document (if applicable) 11. Information Disclosure Statement (IDS) Form PTO-1449 Copies of References 12. Preliminary Amendment
i. Deletion of Inventor(s) Signed statement attached deleting inventor(s) named in the prior application	13. ☐ Return Receipt Postcard (Should be specifically itemized) 14. ☐ Small Entity Statement(s) ☐ Enclosed
 Incorporation by Reference: The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein. Microfiche Computer Program 	Statement filed in prior application; status still proper and desired 15. Certified Copy of Priority Document(s) 16. Other: Appendix A, Appendix B, Appendix C, Appendix D, Appendix E
 7. Nucleotide and/or Amino Acid Sequence Submission a. Computer Readable Copy b. Paper Copy c. Statement verifying above copies 	
	propriate box and supply the requisite information in
amend the specification by inserting the "This is a ☐ continuation ☐ divisional or ☐ Serial No. , filed on	; Group Art Unit: 35 USC §120. The Commissioner is requested to following sentence before the first line:
☐ International Application, filed on	, and which designates the 0.3.

APPLICATION FEES					
Basic Fee					\$690.00
CLAIMS	Nui	MBER FILED	Number Extra	RATE	
Total Claims	3	-20=	0	x \$18.00	\$0.00
Independent Claims	3	- 3=	0	x \$78.00	\$0.00
☐ Multiple Dependent Claims(s) if applicable +\$260.00			+\$260.00	\$	
<u> </u>			Total of above	e calculations =	\$690.00
		R	eduction by 50% for filing b	y small entity =	\$()
Assignment fee if applicable + \$40.00					\$
¥				TOTAL =	\$690.00

Sing and gree gues gree 1.7 ťħ Hard I work in North

	FAPPLICATION TRANSMITTAL		Attorney Docket No. 204843				
19. 🛛 Please	9. Please charge my Deposit Account No. 12-1216 in the amount of \$690.00.						
20.	0. A check in the amount of \$ is enclosed.						
following type a. 🛛 F	=						
future r extens specific	The Commissioner is hereby generally authorized under 37 CFR §1.136(a)(3) to treat any future reply in this or any related application filed pursuant to 37 CFR §1.53 requiring an extension of time as incorporating a request therefor, and the Commissioner is hereby specifically authorized to charge Deposit Account No. 12-1216 for any fee that may be due in connection with such a request for an extension of time.						
	23. CORRESPON						
Customer Number: 23460 , Reg. No. Leydig, Voit & Mayer, Ltd. Two Prudential Plaza, Suite 4900 180 North Stetson Chicago, Illinois 60601-6780 (312) 616-5600 (telephone)							
Name	(312) 010-3700 (lacsimile)						
Signature // Lill							
Date	April 24, 2000						

Certificate of Mailing Under 37 CFR §1.10

I hereby certify that this Utility Patent Application Transmittal and all accompanying documents are being deposited with the United States Postal Service "Express Mail Post Office To Addressee" Service under 37 CFR §1.10 on the date indicated below and is addressed to: Assistant Commissioner for Patents, Box Patent Application, Washington, D.C. 20231.

Matthew W. Olson	Mitte G-Osy	April 24, 2000
Name of Person Signing	Signature	Date

UTILITY (Rev. 3/13/2000)

PATENT APPLICATION

Invention Title:

EXPOSING BLUETOOTH COMPLIANT WIRELESS DEVICE CONNECTIONS AS MODEMS OR SOCKETS

Inventors:			
GILIBERTO, Louis			
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
ADERMAN, Stan			
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
HOLAN, Dolon			
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
ROUKBI, Husni			
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
BERTOGLIO, Mark			
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
JOY, Joseph			
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
ZINTEL, Michael			
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY
Murching, Arvind			
INVENTOR'S NAME	CITIZENSHIP	CITY OF RESIDENCE	STATE or FOREIGN COUNTRY

Be it known that the inventors listed above have invented a certain new and useful invention with the title shown above of which the following is a specification.

10

15

20

25

EXPOSING BLUETOOTH COMPLIANT WIRELESS DEVICE CONNECTIONS AS MODEMS OR SOCKETS

TECHNICAL FIELD

This invention relates generally to wireless interface technology and, more particularly, relates to the interface between computer software applications and wireless devices operating in accordance with the Bluetooth specification.

BACKGROUND OF THE INVENTION

To provide the greatest compatibility between software and hardware components on a computer system, the operating system of the computer defines certain interfaces which can be accessed and used by the programmers of the software components and which are to be provided and supported by the designers of hardware components. Thus, by using the defined interface, the software component can be assured of compatibility with all of the hardware components which support the interface. Similarly, a hardware component providing a specific interface can be assured that software components will be able to locate and access the functionality provided by the hardware component through the interface.

Generally, computers and other electronic devices are interconnected via physical cables or wires. These communication paths allow for the exchange of data or control information between such devices. However, it is increasingly recognized that certain advantages arise from the elimination of cables and wires to interconnect devices. Such advantages include ease of configuration and reconfiguration, due to the elimination of the need to physically add, remove, or displace a physical medium. Furthermore, space

10

15

20

which would traditionally be used for device interconnection media may be given to other uses. Furthermore, device mobility is increased through the use of wireless connections.

One method for providing wireless connections between devices employs a light wave in the Infrared region of the electromagnetic spectrum to link devices. The IrDA (Infrared Data Association) protocol defines one such connection mechanism.

Unfortunately, such a mechanism must usually operate in a line of sight manner. That is to say that any opaque obstruction between transmitter and receiver will prevent proper operation. Additionally, IR transmitters are typically not omnidirectional when incorporated into a communicating device, so that for proper operation, the transmitter must be pointed generally in the direction of the receiver, within some nominal deviation such as 30 degrees. Finally, IR transmitters are typically fairly low power devices, and accordingly the range of IR links is usually limited to approximately one meter.

Radio frequency links solve many of the problems inherent in Infrared links, however, a radio frequency connection scheme is needed whereby a variety of applications can easily access the radio link through a connection mechanism that provides an appropriate interface. One protocol which defines communication between wireless devices through radio frequency links is the Bluetooth specification. Bluetooth devices do not require a line of sight with one another to operate, and their range can be significantly greater than that of IR links. However, one difficulty with the Bluetooth specification is that very few computer software programs are written to communicate with Bluetooth compliant devices. Another difficulty with the Bluetooth specification is that Bluetooth compliant devices are presented to computer software programs as serial interfaces. There are be numerous situations it which such a serial presentation can be

10

15

20

inefficient or even confusing for certain types of computer software applications, such as simple networking applications. Yet another difficulty with the Bluetooth specification is that, while it supports up to 30 emulated RS-232 ports, computer software programs are generally required to know how to communicate through such an emulated port in a device-specific manner.

SUMMARY OF THE INVENTION

Accordingly, the present invention provides a method and computer program product for providing an interface to a Bluetooth compliant device which can emulate a modem such that computer software programs can communicate through the Bluetooth compliant device in the same manner in which they would communicate through a standard modem to access a dial-up, wide area network.

The present invention also provides a method and computer program product for providing an interface to a Bluetooth compliant device which can emulate a network socket such that computer software programs can communicate through the Bluetooth compliant device seemingly in the same manner in which they would communicate through a standard network interface card to access a local area network.

Additionally, the present invention provides a method by which the interface to a Bluetooth compliant device is dependent on the nature of the Bluetooth compliant device.

Additional features and advantages of the invention will be made apparent from the following detailed description of illustrative embodiments which proceeds with reference to the accompanying figures.

10

15

20

BRIEF DESCRIPTION OF THE DRAWINGS

While the appended claims set forth the features of the present invention with particularity, the invention, together with its objects and advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

Figure 1 is a block diagram generally illustrating an exemplary computer system on which the present invention resides;

Figure 2 is a block diagram generally illustrating a seven-layer network model;

Figure 3 is an architectural diagram of various system components used in an embodiment of the invention; and

Figure 4 is a flow chart illustrating steps taken in an embodiment of the invention to interface applications of different types to a radio frequency link.

DETAILED DESCRIPTION OF THE INVENTION

Turning to the drawings, wherein like reference numerals refer to like elements, the invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor based or programmable consumer electronics,

10

15

20

network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

With reference to Fig. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional personal computer 20, including a processing unit 21, a system memory 22, and a system bus 23 that couples various system components including the system memory to the processing unit 21. The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system (BIOS) 26, containing the basic routines that help to transfer information between elements within the personal computer 20, such as during start-up, is stored in ROM 24. The personal computer 20 further includes a hard disk drive 27 for reading from and writing to a hard disk 60, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to a removable optical disk 31 such as a CD ROM or other optical media.

The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical disk drive interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable

10

15

20

instructions, data structures, program modules and other data for the personal computer 20. Although the exemplary environment described herein employs a hard disk 60, a removable magnetic disk 29, and a removable optical disk 31, it will be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories, read only memories, and the like may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk 60, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more applications programs 36, other program modules 37, and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and a pointing device 42. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 47 or other type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, personal computers typically include other peripheral output devices, not shown, such as speakers and printers.

The personal computer 20 may operate in a networked environment using logical connections to one or more remote computers or devices, such as a remote computer 49 or RF device 64. The remote computer 49 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically

10

15

20

includes many or all of the elements described above relative to the personal computer 20, although only a memory storage device 50 has been illustrated in Fig. 1. The Radio Frequency (RF) device 64 can be a cellular phone, digital camera, another personal computer, or other device which includes the capability to communicate through the RF spectrum. The logical connections depicted in Fig. 1 include a local area network (LAN) 51 and a wide area network (WAN) 52, and an RF connection 63. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the personal computer 20 is connected to the local network 51 through a network interface or adapter 53. When used in a WAN networking environment, the personal computer 20 typically includes a modem 54 or other means for establishing communications over the WAN 52. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. When used in conjunction with an RF connection 63, the personal computer 20 includes an RF interface 62. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

In the description that follows, the invention will be described with reference to acts and symbolic representations of operations that are performed by one or more computers, unless indicated otherwise. As such, it will be understood that such acts and operations, which are at times referred to as being computer-executed, include the

manipulation by the processing unit of the computer of electrical signals representing data in a structured form. This manipulation transforms the data or maintains it at locations in the memory system of the computer, which reconfigures or otherwise alters the operation of the computer in a manner well understood by those skilled in the art. The data structures where data is maintained are physical locations of the memory that have particular properties defined by the format of the data. However, while the invention is being described in the foregoing context, it is not meant to be limiting as those of skill in the art will appreciate that various of the acts and operations described hereinafter may also be implemented in hardware.

In accordance with the invention, and turning to Figure 2, the Open Systems

Interconnection (OSI) seven-layer model is shown. This model is an industry standard abstraction of computer networking. The application layer 100 directly serves the end user and supports the software applications with which the user interacts. The presentation layer 102 provides the mechanisms which interpret data being sent from the application layer 100 on one computer to the application layer on another. The session layer 104 describes the organization of the data being transferred. The transport layer 106 acts as a final error correcting layer to ensure the data is delivered accurately, in the proper sequence, and with no loss or duplication. The network layer 108 defines the addressing and routing of the data across the network. It controls the operation of the local sub-network and decides which physical path the data should take, given network conditions, priority of service, and other factors. The data link layer 110 controls the transmission of blocks of data, or packets, across the network, and provides more

fundamental error correction. The data link layer 110 is divided into two sublayers: the

20

5

10

15

10

15

20

logical link control (LLC) sublayer and the media access control (MAC) sublayer. The LLC sublayer ensures error-free transmission of data frames by maintaining logical links, controlling frame flow, sequencing frames, acknowledging frames, and retransmitting unacknowledged frames. The MAC sublayer manages access to the network, checks frame errors and address recognition of the received frames. Protocols which include an LLC sublayer need only a minimal transport layer 106. Finally, the physical layer 112 carries the signals which are sent to the network connection 114. Generally, the physical layer 112 is implemented in the hardware connecting the computer 20 to the network connection 114.

A Network Device Interface Specification (NDIS) 116 can reside between the network layer 108 and the data link layer 110. The NDIS 116 can provide a library of interfaces between the software components and the hardware components. The NDIS 116 can define a fully abstracted environment for network interface card (NIC) driver development by providing routines for every external function that a NIC driver would need to perform. Thus, the NDIS 116 can provide interfaces for communication between a NIC driver and a overlying protocol driver and between a NIC driver and the underlying NIC hardware itself.

Generally the application layer 100, presentation layer 102, session layer 104, transport layer 106, and the network layer 108 are implemented in software components operating on a computer. The data link layer 110 and the physical layer 112 are generally implemented by the hardware components, such as a network interface card. The NDIS 116 library can be used by a software driver implemented in the transport layer 110 to communicate with a network interface card driver implemented at the data link layer 110.

10

15

20

A transport layer driver generally implements a network protocol stack, such as the well known Transfer Control Protocol / Internet Protocol (TCP/IP) stack used on the Internet. If the transport layer software driver has a packet of data to be transmitted, it can call the NIC driver by means of an interface from the NDIS 116 library, and pass down the packet to be transmitted. Similarly, the NIC driver can use an interface of the NDIS 116 to pass the packet to the NIC itself for transmission across the network. The NDIS 116 interface can call the operating system specific components which perform the transmission at the NIC. The NDIS 116 interfaces can also be used by the NIC driver to communicate with the transport layer software driver and pass up a received packet of data, or other information.

According to the "Specification of the Bluetooth System" Version 1.0B (December 1,1999), incorporated herein by reference in its entirety, RFCOMM supports up to 30 emulated RS-232 (COM) port connections between any two devices. See also the "Windows Wireless Architecture" presentation at Appendix B, the "Bluetooth Architecture Overview" presentation at Appendix C, the "Bluetooth Experience in Windows" presentation at Appendix D, and the "Bluetooth Stack in Windows" presentation at Appendix E. However, Dial-Up Networking (DUN) connections provide specific services that are best presented as a modem. Accordingly, when a DUN profile is exposed as a COM port rather than as a modem connection, the relevant client application must have the ability to communicate in a device-specific way with a device.

In keeping with an embodiment of the invention, DUN services are exposed by RFCOMM to the application as a modem connection, allowing the client application to use standard Telephony API (TAPI) and Unimodem interfaces. Thus applications and

services which are not specifically adapted for use within the Bluetooth protocol can nonetheless utilize a communications device as a standard communications device, hiding the implementation-specific differences between Bluetooth and Dial-up Networking connections.

RFCOMM is implemented as described in the "Specification of the Bluetooth System" Version 1.0B Part F1 entitled "RFCOMM with TS 07.10 incorporated herein by reference in its entirety and attached at Appendix A, with certain changes to effect the desired functionality. The following components, most of which appear in the architectural diagram of Fig. 3, are used to expose a Bluetooth RFCOMM Dial-Up Networking connection as a modem rather than as a COM port: RFCOMM.SYS (the Bluetooth RFCOMM driver) 301; BTHPORT.SYS (the Bluetooth port driver implementing L2CAP and HCI.) 303; TDI (transport device interface) 305; PnP (the "Plug'N'Play" system); BTHMDM.SYS (the Bluetooth modem driver) 307; and MODEM.SYS (the Unimodem driver) 309. As one of skill in the art will know, the Plug'N'Play system is a combination of hardware and software support that enables a computer system to recognize and adapt to hardware configuration changes with little or no user intervention.

When a new device conforming to the Bluetooth specification is detected by the computer, BTHPORT.SYS enumerates the new device as a Physical Device Object (PDO). As is known by those of skill in the art, a PDO represents the whole range of functionality available to a component. RFCOMM is alerted to this new device by way of BTHPORT.SYS. RFCOMM.SYS is loaded as the Functional Device Object (FDO)

10

15

20

by the Plug'N'Play system (PnP). As is also known by those of skill in the art, an FDO represents a set of functions of device available to a function driver.

RFCOMM examines the Service Discovery Protocol (SDP) database of the remote RFCOMM device. If the remote device is a DUN device, RFCOMM enumerates a new PDO. For Bluetooth LAN access points and PC's acting as LAN access points, RFCOMM.SYS enumerates a PDO that will load an instance of a Null modem device in BTHMDM.SYS. For Bluetooth modems acting as a Gateway (GW), RFCOMM.SYS will enumerate a PDO that will load an instance of a modem device in BTHMDM.SYS. BTHMDM.SYS communicates to RFCOMM using IO request packets (IRP's) via the TDI interface. Alternatively, BTHMDM.SYS may communicate with IRP's that are not restricted to TDI requests, but that are still Windows Driver Model (WDM) requests.

BTHMDM.SYS exposes an interface to MODEM.SYS that provides a functional equivalent of the standard Windows 2000 SERIAL.SYS driver. This has the effect of emulating an RS-232 modem connection from the viewpoint of MODEM.SYS.

MODEM.SYS then provides a Unimodem interface to Unimodem clients such as TAPI, allowing the device to be addressed as a standard communications device, e.g. a modem.

To permit peer-to-peer DUN communications between two PC's, it is preferable that one of the PC's acts as a server. The server PC preferably populates its SDP database with and appropriate DUN entry so that the client can identify and connect with it. This is preferably performed at the time that the RFCOMM driver loads, either at system start up, or at the time that the Bluetooth device is connected to the system. RFCOMM.SYS will automatically generate a PDO to represent the server channel to BTHMDM.SYS,

10

15

20

such that the server software may be initialized and ready to handle an incoming connection request.

When a client attempts to connect to the DUN server, BTHPORT.SYS will generate a PDO to represent the new connection. RFCOMM will create an FDO and associate it with the new PDO. However, instead of generating a new PDO for the modem driver, RFCOMM will associate the new PDO and FDO with the already existing PDO being handled by the DUN server.

The communication mechanism described above with reference to Bluetooth DUN connections also applies to dependant profiles such as the LAN access profile and the FAX profile.

According to another embodiment of the invention, RFCOMM is alerted to a new connection by BTHPORT.SYS, as was described in more detail above. If, after examining the SDP database of the new connection, as described above, RFCOMM determines that the connection is not a dial-up networking device, RFCOMM will allow access to the device through the RFCOMM TDI interface. As is known by those of skill in the art, the TDI is the interface which allows higher level components access to the transport layer, which was described in more detail above. This access is extended to the user mode level by AFD.SYS. It is AFD.SYS which provides this access to WinSock. In such a manner, the new device is assigned its own socket and treated in a manner analogous to a network card.

Each transport protocol defines an address that describes endpoint information associated with address objects that are open in the transport layer. The TDI address of an RFCOMM endpoint can be defined as follows:

20

25

Where the TDI_ADDRESS_BLUETOOTH type defines an RFCOMM channel number of the endpoint, if it is supported by the RFCOMM stack. Alternatively, ChannelNumber can specify an L2CAP channel number.

As described above, the TDI allows higher level components to access the transport layer. Such a higher level component is known as a client of the TDI. A client of the TDI can open a server endpoint for accepting incoming connections by any number of methods. One method for doing so is to include a TDI_ADDRESS_BLUETOOTH with a ChannelNumber of either BTH ANY CHANNEL NUMBER or an RFCOMM channel number between one and thirty. Specifying BTH_ANY_CHANNEL_NUMBER will let RFCOMM select an unused channel number. If the client manually selects a channel number and it is in use, the opening of the address object will fail with TDI ADDR IN USE. BdAddr can be set to zero. Another method of opening a server endpoint for accepting incoming connections is to open a connection object and associate it with the above address object. This will be the connection object for the first incoming connection. Because RFCOMM can limit a server channel to just one connection per remote device it is generally not necessary to create a backlog of connection objects for incoming connections. After accepting an incoming connection, the TDI client can simply create another connection object and associate it with the above address object for the next incoming connection.

10

15

20

A TDI client can also open a connection to a remote device through a number of methods. One method for doing so is to open an address object, including a TDI_ADDRESS_BLUETOOTH with a ChannelNumber set to zero. A ChannelNumber of zero indicates that the address object will be used for an outbound connection and the stack will not reserve a server channel number for it. BdAddr can be set to zero. Another method of opening a connection to a remote device is to open a connection object and associate it with the address object. Yet another method of opening a connection to a remote device is to issue a TDI connect request IRP on the connection object. The IRP will contain a TDI_ADDRESS_BLUETOOTH with the BdAddr of the remote device and the destination ChannelNumber.

Existing implementations of the Bluetooth specification map remote devices to a generic serial-type device. Unfortunately, proper configuration of such a system requires user knowledge regarding serial port technology. In an embodiment of the present invention, RFCOMM connections of a particular type are automatically routed to an appropriate corresponding device type within the Microsoft brand WINDOWS operating system using the SDP. Broadly, if a device is not a DUN device, then RFCOMM will allow access to that device through the TDI interface. This access is extended to user mode by AFD.SYS (standard Winsock service provider for transports). In order to allow TDI's addressing model to multiplex multiple connections to the same RFCOMM channel on different RFCOMM sessions, the RFCOMM channel number/remote Bluetooth address pair of the endpoint uniquely identifies each RFCOMM connection. In contrast to the DUN profile, Winsock and AFD will be required to create device objects and handles in a mechanism consistent with existing TDI applications.

10

15

In greater detail, with reference to step 201 of Fig. 4, BTHPORT.SYS initially enumerates a new device PDO. Subsequently, in step 203, RFCOMM.SYS is loaded as the functional driver for this PDO. In step 205, RFCOMM.SYS examines the SDP data base of the remote device to determine device type. If in step 207 it is determined that the device is a DUN device, then in step 211 RFCOMM.SYS enumerates a PDO that has MODEM.SYS as its functional driver and BTHMDM.SYS as a lower filter driver. Otherwise, in step 209, the device is exposed to an API such as Windows Sockets via the TDI interface.

All of the references cited herein, including patents, patent applications, and publications, are hereby incorporated in their entireties by reference.

In view of the many possible embodiments to which the principles of this invention may be applied, it should be recognized that the embodiment described herein with respect to the drawing figures is meant to be illustrative only and should not be taken as limiting the scope of invention. For example, those of skill in the art will recognize that the elements of the illustrated embodiment shown in software may be implemented in hardware and vice versa or that the illustrated embodiment can be modified in arrangement and detail without departing from the spirit of the invention. Therefore, the invention as described herein contemplates all such embodiments as may come within the scope of the following claims and equivalents thereof.

CLAIMS

We claim:

5

10

1. For use in a computer, a method of exposing a dial-up networking device to an application as a modem via RFCOMM, the method comprising the steps of;

detecting a new connection to a remote device;

determining whether the remote device is a dial-up networking device;

if the remote device is the dial-up networking device, enumerating a physical device object corresponding to the remote device; and

using the physical device object to communicate between a Bluetooth modem driver and RFCOMM without requiring the modem driver to specifically utilize an emulated COM port.

- 2. For use in a computer, a method of exposing a remote device to an application as sockets via RFCOMM, the method comprising the steps of;
- detecting a new connection to the remote device;

determining whether the remote device is a dial-up networking device; and if the remote device is not the dial-up networking device, allowing access to the remote device through an interface available to the application; wherein the access to the remote device is established by a RFCOMM channel number and remote device

20 address pair.

3. A method of automatically routing an RFCOMM connection to a proper device type comprising the steps of:

detecting a new device for connection;

determining the type of the new device;

enumerating a physical device object associated with the new device if the new device is a dial-up networking device; and

exposing the device to an application by way of a transport driver interface if the device is not a DUN device.

10

ABSTRACT OF THE INVENTION

A method for providing an interface to a Bluetooth compliant device can emulate a modem such that computer software programs can communicate through the Bluetooth compliant device in the same manner in which they would communicate through a standard modem to access a dial-up, wide area network. The method also supports an interface to a Bluetooth compliant device which can emulate a network socket such that computer software programs can communicate through the Bluetooth compliant device seemingly in the same manner in which they would communicate through a standard network interface card to access a local area network. The method also allows for the interface to a Bluetooth compliant device to be dependent on the nature of the Bluetooth compliant device.

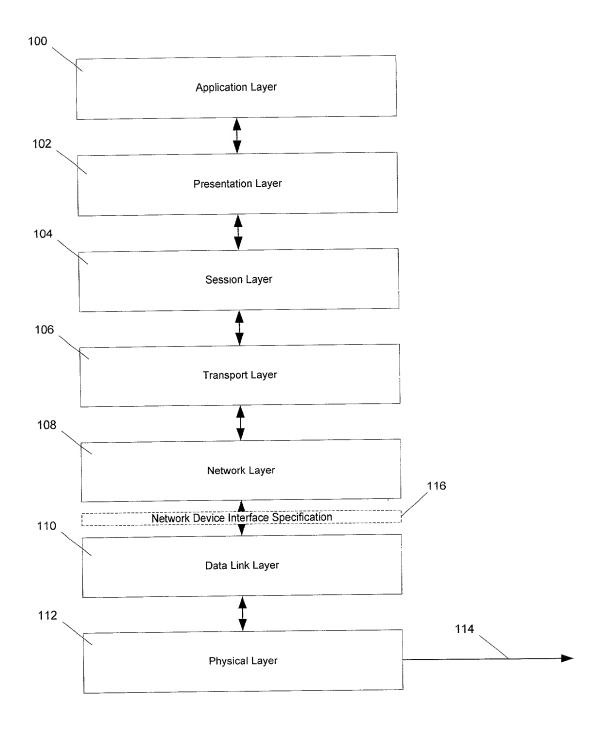


Figure 2

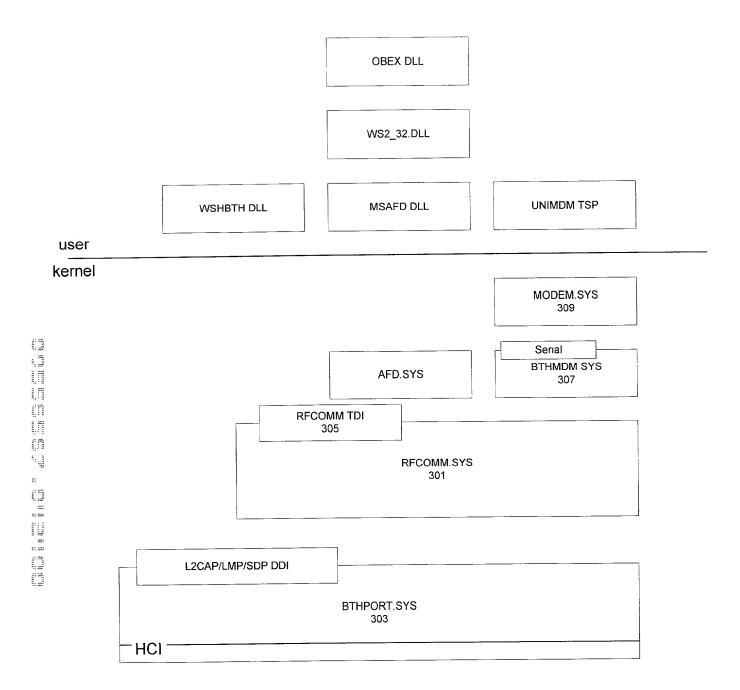


FIG. 3

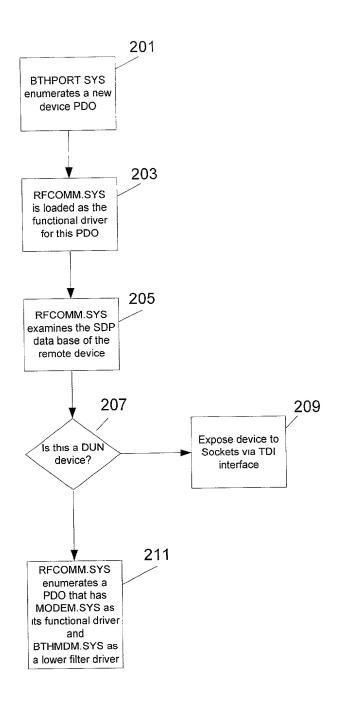
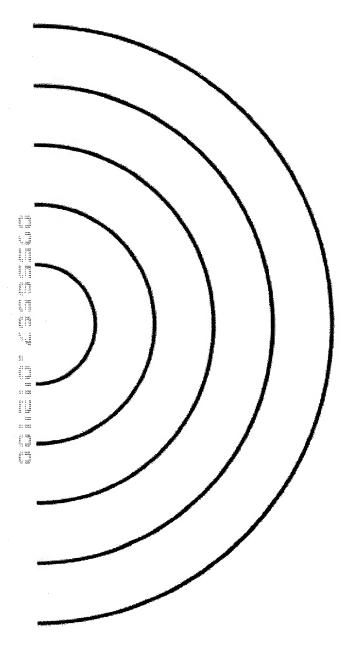


FIG. 4



Specification of the Bluetooth System



Wireless connections made easy

Core

Bluetooth.

v1.0 B December 1st 1999

BLUETOOTH DOC Oil Dec 99 Cate / Day-Month-Year N.S. Document No. 1.C.47/1.0 B	
Responsible c-mell address	

Specification of the Bluetooth System

Version 1.0 B

Revision History

The Revision History is shown in Appendix I on page 868

Contributors

The persons who contributed to this specification are listed in Appendix II on page 879.

Web Site

This specification can also be found on the Bluetooth website: http://www.bluetooth.com

Disclaimer and copyright notice

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

Copyright © 1999

Telefonaktiebolaget LM Ericsson, International Business Machines Corporation, Intel Corporation, Nokia Corporation, Toshiba Corporation.

*Third-party brands and names are the property of their respective owners.

MASTER TABLE OF CONTENTS

For the Bluetooth Profiles, See Volume 2.

Part A			Volume 1 (1:2)	
ADIO SPECIFICATION				
Conte	nts	***************************************	[A] 17	
	1	Scope	[A] 18	
	2	Frequency Bands and Channel Arrangement	[A] 19	
	3	Transmitter Characteristics	[A] 20	
	4	Receiver Characteristics	[A] 24	
	5	Appendix A	[A] 28	
	6	Appendix B	[A] 31	
Part B			Volume 1 (1:2)	
BASEBAND	SP	ECIFICATION		
Conte	nts		[B] 35	
	1	General Description	[B] 41	
	2	Physical Channel	[B] 43	
	3	Physical Links	[B] 45	
	4	Packets	[B] 47	
	5	Error Correction	[B] 67	
	6	Logical Channels	[B] 77	
	7	Data Whitening	[B] 79	
	8	Transmit/Receive Routines	[B] 81	
	9	Transmit/Receive Timing	[B] 87	
	10	Channel Control	[B] 95	
	11	Hop Selection	[B] 127	
	12	Bluetooth Audio	[B] 139	
	13	Bluetooth Addressing	[B] 143	
	14	Bluetooth Security	[B] 149	
	15	List of Figures	[B] 179	
	16	List of Tables	[B] 183	

Part C			Volume 1 (1:2)
LINK MANAG	EF	PROTOCOL	
Conte	nts		[C] 187
•	1	General	[C] 191
2	2	Format of LMP	[C] 192
;	3	The Procedure Rules and PDUs	[c] 193
4	4	Connection Establishment	[C] 225
	5	Summary of PDUs	[C] 226
(6	Test Modes	[C] 237
•	7	Error Handling	[C] 239
;	8	List of Figures	[C] 241
!	9	List of Tables	[C] 243
Part D			Volume 1 (1:2)
SPECIFICAT	101	CONTROL AND ADAPTATION PROT	
	1	Introduction	
	2	General Operation	
	3	State Machine	
	4	Data Packet Format	
	5	Signalling	[D] 275
	6	Configuration Parameter Options	
	7	Service Primitives	[D] 295
	8	Summary	[D] 313
	9	References	
	10	List of Figures	[D] 315
	11	List of Tables	[D] 316
	Ter	ms and Abbreviations	
		pendix A: Configuration MSCs	
	Аp	pendix B: Implementation Guidelines	[D] 321

Part E			Volume 1 (1:2)
SERVICE I	DISC	OVERY PROTOCOL (SDP)	
Con	tents	s	[E] 325
	1	Introduction	[E] 327
	2	Overview	[E] 330
	3	Data Representation	[E] 341
	4	Protocol Description	[E] 344
	5	Service Attribute Definitions	[E] 358
	Ap	pendix A- Background Information	[E] 370
	Ap	pendix B – Example SDP Transactions	[E] 371
Part F:1			Volume 1 (1:2)
RFCOMM			
Cor	ntent	s	
	1	Introduction	
	2	RFCOMM Service Overview	
	3	Service Interface Description	
	4	TS 07.10 Subset Supported by RFCOMM	[F:1] 396
	5	TS 07.10 Adaptations for RFCOMM	[F:1] 398
	6	Flow Control	[F:1] 403
	7	Interaction with Other Entities	[F:1] 405
	8	References	[F:1] 408
	9	Terms and Abbreviations	[F:1] 409
Part F:2			Volume 1 (1:2)
		PERABILITY	
Co	ntent	is	
	1	Introduction	
	2	OBEX Object and Protocol	
	3	OBEX over RFCOMM	
	4	OBEX over TCP/IP	
	5	Bluetooth Application Profiles using OBEX.	
	6	References	[F:2] 427
	7	List of Acronyms and Abbreviations	[F:2] 428

Part F:3			Volume 1 (1:2)
TELEPHON	IY C	ONTROL PROTOCOL SPECIFICATION	
Cont	tents		[F:3] 431
	1	General Description	[F:3] 435
	2	Call Control (CC)	[F:3] 439
	3	Group Management (GM)	[F:3] 449
	4	Connectionless TCS (CL)	[F:3] 455
	5	Supplementary Services (SS)	[F:3] 456
	6	Message formats	[F:3] 459
	7	Message coding	[F:3] 471
	8	Message Error handling	
	9	Protocol Parameters	[F:3] 489
	10	References	[F:3] 490
	11	List of Figures	[F:3] 491
	12	List of Tables	[F:3] 492
	Ap	pendix 1 - TCS Call States	[F:3] 493
Part F:4			Volume 1 (1:2
INTEROPE	RAB	ILITY REQUIREMENTS FOR BLUETOOTH AS	S A WAP BEAREF
Con	tent	s	[F:4] 497
	1	Introduction	[F:4] 499
	2	The Use of WAP In the Bluetooth Environment	ent[F:4] 500
	3	WAP Services Overview	[F:4] 502
	4	WAP in the Bluetooth Piconet	[F:4] 506
	5	Interoperability Requirements	[F:4] 51 ⁻
	6	Service Discovery	
	7	References	

Part H:1			Volume 1 (2:2)
HOST CONT	rRC	LLER INTERFACE FUNCTIONAL SPECIFIC	ATION
Conte	ents	3	[H:1] 519
	1	Introduction	[H:1] 524
	2	Overview of Host Controller Transport Layer.	[H:1] 528
	3	HCI Flow Control	[H:1] 529
	4	HCI Commands	[H:1] 531
	5	Events	[H:1] 703
	6	List of Error Codes	[H:1] 745
	7	List of Acronyms and Abbreviations	[H:1] 755
	8	List of Figures	[H:1] 756
	9	List of Tables	[H:1] 757
Part H:2			Volume 1 (2:2)
HCI USB TR	RAN	ISPORT LAYER	-
Conte	ent	S	[H:2] 761
	1	Overview	[H:2] 762
	2	USB Endpoint Expectations	[H:2] 764
	3	Class Code	[H:2] 771
	4	Device Firmware Upgrade	[H:2] 772
	5	Limitations	[H:2] 773
Part H:3			Volume 1 (2:2)
HCI RS232	TR	ANSPORT LAYER	
Cont	ent	s	[H:3] 777
	1	General	[H:3] 778
	2	Overview	[H:3] 779
	3	Negotiation Protocol	[H:3] 780
	4	Packet Transfer Protocol	[H:3] 784
	5	Using delimiters with COBS for synchronizat	ion[H:3] 785
	6	Using RTS/CTS for Synchronization	[H:3] 788
	7	References	[H:3] 794

Part H:4		Volu	me 1 (2:2)
HCI UART T	RAN	NSPORT LAYER	
Conte	ents		.[H:4] 797
	1	General	. [H:4] 798
	2	Protocol	. [H:4] 799
	3	RS232 Settings	. [H:4] 800
	4	Error Recovery	. [H:4] 801
Part I:1		Volu	me 1 (2:2)
BLUETOOT	нт	EST MODE	
Cont	ents	s	[I:1] 805
	1	General Description	[l:1] 806
	2	Test Scenarios	[l:1] 808
	3	Outline of Proposed LMP Messages	[l:1] 817
	4	References	[l:1] 819
Part I:2		Volu	me 1 (2:2)
BLUETOOT	TH C	OMPLIANCE REQUIREMENTS	
Con	tent	s	[l:2] 823
	1	Scope	[l:2] 825
	2	Terms Used	[I:2] 826
	3	Legal Aspects	[I:2] 828
	4	The Value of the Bluetooth Brand	[1:2] 829
	5	The Bluetooth Qualification Program	[I:2] 830
	6	Bluetooth License Requirements for Products	[1:2] 832
	7	Bluetooth License Provisions for Early Products	[l:2] 836
	8	Bluetooth Brand License Provisions for Special Products & Marketing	[l:2] 837
	9	Recommendations Concerning Information about a Product's Bluetooth Capabilities	[l:2] 838
	10	Quality Management, Configuration Management and Version Control	[i:2] 839
	11	Appendix A – Example of a "Bluetooth Capability Statement"	[l:2] 840
	12	Appendix B - Marketing Names of Bluetooth Profile	es.[l:2] 841

Volume 1 (2:2)

Volume 1 (2:2)

Bluetooth. Volume 1 (2:2) Part I:3 **TEST CONTROL INTERFACE** Contents[1:3] 845 Introduction[1:3] 847 2 General Description[1:3] 849 3 Test Configurations[I:3] 854 TCI-L2CAP Specification[1:3] 856 Abbreviations[1:3] 866 Profiles - see Volume 2 Part K Volume 1 (2:2) Appendix I Appendix II Volume 1 (2:2)

CONTRIBUTORS [@:II] 881

ACRONYMS AND ABBREVIATIONS......[@:III] 891

SAMPLE DATA

Appendix IV

Appendix III

Contents		[@:IV] 901
1	Encryption Sample Data	[@:IV] 902
2	Frequency Hopping Sample Data—Mandatory Scheme	[@:IV] 937
3	Access Code Sample Data	
4	HEC and Packet Header Sample Data	[@:IV] 953
5	CRC Sample Data	[@:IV] 954
6	Complete Sample Packets	[@:IV] 955
7	Whitening Sequence Sample Data	[@:IV] 957
8	FEC Sample Data	[@:IV] 960
9	Encryption Key Sample Data	[@:IV] 961

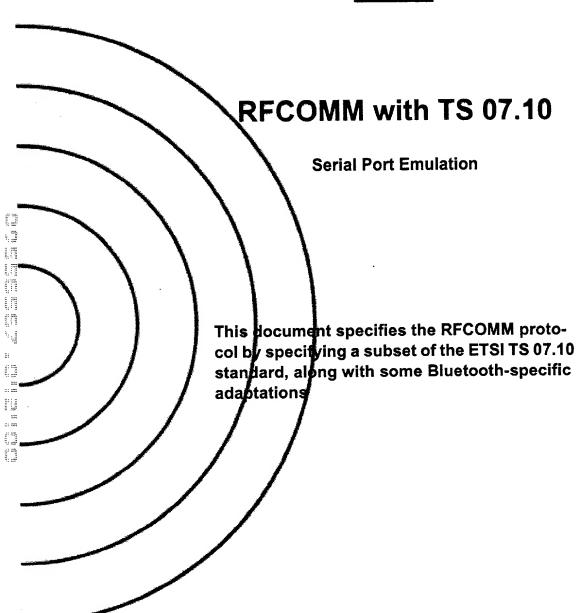
Bluetooth. Volume 1 (2:2) Appendix V **BLUETOOTH AUDIO** Contents[@:V] 987 General Audio Recommendations.....[@:V] 989 Volume 1 (2:2) Appendix VI **BASEBAND TIMERS** Contents[@:VI] 995 Baseband Timers......[@:VI] 996 Volume 1 (2:2) Appendix VII **OPTIONAL PAGING SCHEMES** Contents[@:VII] 1001 General[@:VII] 1003 Optional Paging Scheme I[@:VII] 1004 Volume 1 (2:2) Appendix VIII **BLUETOOTH ASSIGNED NUMBERS** Contents[@:VIII] 1011 Bluetooth Baseband[@:VIII] 1012 Link Manager Protocol (LMP)[@:VIII] 1018 Logical Link Control and Adaptation Protocol.....[@:VIII] 1019 Service Discovery Protocol (SDP)[@:VIII] 1020 References[@:VIII] 1028 Terms and Abbreviations[@:VIII] 1029 List of Figures[@:VIII] 1030 7 List of Tables[@:VIII] 1031

Bluetooth.

Appendix IX		Volume 1	(2:2)				
MESSAGE SEQ	MESSAGE SEQUENCE CHARTS						
Contents		[@:IX]	1035				
1	Introduction	[@:IX]	1037				
2	Services Without Connection Request	[@:IX]	1038				
3	ACL Connection Establishment and Detach	ment.[@:IX]	1042				
4	Optional Activities After ACL Connection Establishment	[@:IX]	1050				
5	SCO Connection Establishment and Detach	ment [@:IX]	1059				
6	Special Modes: Sniff, Hold, Park	[@:IX]	1062				
7	Buffer Management, Flow Control	[@:IX]	1068				
8	Loopback Mode	[@: X]	1070				
9	List of Acronyms and Abbreviations	[@:IX]	1073				
10	List of Figures	[@:IX]	1074				
11	List of Tables	[@: X]	1075				
12	References	[@:IX]	1076				
Alphabetical In	dex		1077				

13

Part F:1



Bluetooth.

Bluetooth.

CONTENTS

1	Intro	duction	389
	1.1	Overview	
	1.2	Device Types	
	1.3	Byte Ordering	390
2	RFC	OMM Service Overview	391
	2.1	RS-232 Control Signals	
	2.2	Null Modem Emulation	
	2.3	Multiple Emulated Serial Ports	393 s .393
		2.3.2 Multiple Emulated Serial Ports and Multiple BT Devices	393
3	Serv	ice Interface Description	395
	3.1	Service Definition Model	
4		7.10 Subset Supported by RFCOMM	
4	4.1	Options and Modes	396
	4.2	Frame Types	
	4.3	Commands	
	4.4	Convergence Layers	
5	TS 0	7.10 Adaptations for RFCOMM	
-	5.1	Media Adaptation	398
		5.1.1 FCS calculation	398
	5.2	TS 07.10 Multiplexer Start-up and Closedown Procedure 5.2.1 Start-up procedure	399 399
		5.2.2 Close-down procedure	399
		5.2.3 Link loss handling	
	5.3	System Parameters	
	5.4	DLCI allocation with RFCOMM server channels	
	5.5	Multiplexer Control Commands	401
		5.5.1 Remote Port Negotiation Command (RPN)	401
		5.5.2 Remote Line Status Command (RLS)	402
		5.5.3 DLC parameter negotiation (PN)	402
6	Flow	v Control	403
	6.1	L2CAP Flow Control in Overview	
	6.2	Wired Serial Port Flow Control	
	6.3	RFCOMM Flow Control	403
	6.4	Port Emulation Entity Serial Flow Control	403

RFC	OMM wit	h TS 07.10	Bluetooth.
7	Inter	action with Other Entities	405
	7.1	Port Emulation and Port Proxy Entities	405
		7.1.1 Port Emulation Entity	405
		7.1.2 Port Proxy Entity	405
	7.2	Service Registration and Discovery	405
	7.3		
		7.3.1 Reliability	407
		7.3.2 Low power modes	407
8	Refe	erences	408
9	Tern	ns and Abbreviations	409

Bluetooth.

1 INTRODUCTION

The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. The protocol is based on the ETSI standard TS 07.10. This document does not contain a complete specification. Instead, references are made to the relevant parts of the TS 07.10 standard. Only a subset of the TS 07.10 standard is used, and some adaptations of the protocol are specified in this document.

1.1 OVERVIEW

RFCOMM is a simple transport protocol, with additional provisions for emulating the 9 circuits of RS-232 (EIATIA-232-E) serial ports.

The RFCOMM protocol supports up to 60 simultaneous connections between two BT devices. The number of connections that can be used simultaneously in a BT device is implementation-specific.

1.2 DEVICE TYPES

For the purposes of RFCOMM, a complete communication path involves two applications running on different devices (the communication endpoints) with a communication segment between them. Figure 1.1 shows the complete communication path. (In this context, the term *application* may mean other things than end-user application; e.g. higher layer protocols or other services acting on behalf of end-user applications.)

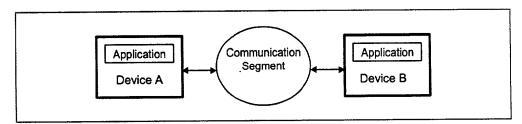


Figure 1.1: RFCOMM Communication Segment

RFCOMM is intended to cover applications that make use of the serial ports of the devices in which they reside. In the simple configuration, the communication segment is a BT link from one device to another (direct connect), see Figure 1.2. Where the communication segment is another network, BT is used for the path between the device and a network connection device like a modem. RFCOMM is only concerned with the connection between the devices in the direct connect case, or between the device and a modem in the network case. RFCOMM can support other configurations, such as modules that communicate via BT on one side and provide a wired interface on the other side, as shown in Figure 1.3. These devices are not really modems but offer a similar service. They are therefore not explicitly discussed here.

Bluetooth.

Basically two device types exist that RFCOMM must accommodate. Type 1 devices are communication end points such as computers and printers. Type 2 devices are those that are part of the communication segment; e.g. modems. Though RFCOMM does not make a distinction between these two device types in the protocol, accommodating both types of devices impacts the RFCOMM protocol.

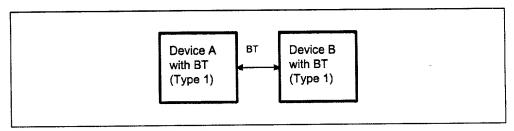


Figure 1.2: RFCOMM Direct Connect

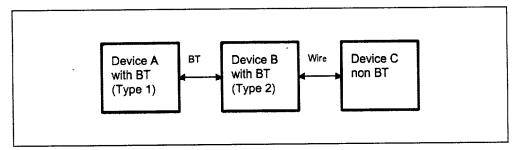


Figure 1.3: RFCOMM used with legacy COMM device

The information transferred between two RFCOMM entities has been defined to support both type 1 and type 2 devices. Some information is only needed by type 2 devices while other information is intended to be used by both. In the protocol, no distinction is made between type 1 and type 2. It is therefore up to the RFCOMM implementers to determine if the information passed in the RFCOMM protocol is of use to the implementation. Since the device is not aware of the type of the other device in the communication path, each must pass on all available information specified by the protocol.

1.3 BYTE ORDERING

This document uses the same byte ordering as the TS 07.10 specification; i.e. all binary numbers are in Least Significant Bit to Most Significant Bit order, reading from left to right.

Bluetooth.

2 RFCOMM SERVICE OVERVIEW

RFCOMM emulates RS-232 (EIATIA-232-E) serial ports. The emulation includes transfer of the state of the non-data circuits. RFCOMM has a built-in scheme for null modem emulation.

In the event that a baud rate is set for a particular port through the RFCOMM service interface, that will not affect the actual data throughput in RFCOMM; i.e. RFCOMM does not incur artificial rate limitation or pacing. However, if either device is a type 2 device (relays data onto other media), or if data pacing is done above the RFCOMM service interface in either or both ends, actual throughput will, on an average, reflect the baud rate setting.

RFCOMM supports emulation of multiple serial ports between two devices and also emulation of serial ports between multiple devices, see Section 2.3 on page 393.

2.1 RS-232 CONTROL SIGNALS

RFCOMM emulates the 9 circuits of an RS-232 interface. The circuits are listed below.

Pin	Gircuit Name
102	Signal Common
103	Transmit Data (TD)
104	Received Data (RD)
105	Request to Send (RTS)
106	Clear to Send (CTS)
107	Data Set Ready (DSR)
108	Data Terminal Ready (DTR)
109	Data Carrier Detect (CD)
125	Ring Indicator (RI)

Table 2.1: Emulated RS-232 circuits in RFCOMM

2.2 NULL MODEM EMULATION

RFCOMM is based on TS 07.10. When it comes to transfer of the states of the non-data circuits, TS 07.10 does not distinguish between DTE and DCE devices. The RS-232 control signals are sent as a number of DTE/DCE independent signals, see Table 2.2.

Bluetooth.

TS 07,10 Signals	Corresponding RS-232 Control Signals
RTC	DSR, DTR
RTR	RTS, CTS
IC	RI
DV THE PRESENT	ist bcp

Table 2.2: TS 07.10 Serial Port Control Signals

The way in which TS 07.10 transfers the RS-232 control signals creates an implicit null modem when two devices of the same kind are connected together. Figure 2.1 shows the null modem that is created when two DTE are connected via RFCOMM. No single null-modem cable wiring scheme works in all cases; however the null modem scheme provided in RFCOMM should work in most cases.

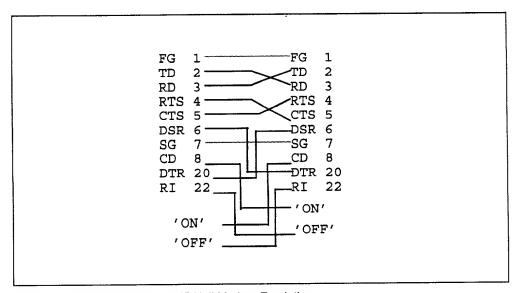


Figure 2.1: RFCOMM DTE-DTE Null Modem Emulation

Bluetooth.

2.3 MULTIPLE EMULATED SERIAL PORTS

2.3.1 Multiple Emulated Serial Ports between two Devices

Two BT devices using RFCOMM in their communication may open multiple emulated serial ports. RFCOMM supports up to 60 open emulated ports; however the number of ports that can be used in a device is implementation-specific.

A Data Link Connection Identifier (DLCI) [1] identifies an ongoing connection between a client and a server application. The DLCI is represented by 6 bits, but its usable value range is 2...61; in TS 07.10, DLCI 0 is the dedicated control channel, DLCI 1 is unusable due to the concept of Server Channels, and DLCI 62-63 is reserved. The DLCI is unique within one RFCOMM session between two devices. (This is explained further in Section 2.3.2) To account for the fact that both client and server applications may reside on both sides of an RFCOMM session, with clients on either side making connections independent of each other, the DLCI value space is divided between the two communicating devices using the concept of RFCOMM server channels. This is further described in Section 5.4.

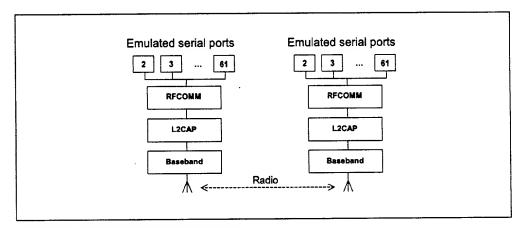


Figure 2.2: Multiple Emulated Serial Ports.

2.3.2 Multiple Emulated Serial Ports and Multiple BT Devices

If a BT device supports multiple emulated serial ports and the connections are allowed to have endpoints in different BT devices, then the RFCOMM entity must be able to run multiple TS 07.10 multiplexer sessions, see Figure 2.3. Note that each multiplexer session is using its own L2CAP channel ID (CID). The ability to run multiple sessions of the TS 07.10 multiplexer is optional for RFCOMM.

Bluetooth.

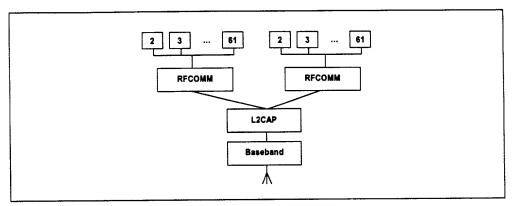


Figure 2.3: Emulating serial ports coming from two BT devices.

Bluetooth.

3 SERVICE INTERFACE DESCRIPTION

RFCOMM is intended to define a protocol that can be used to emulate serial ports. In most systems, RFCOMM will be part of a port driver which includes a serial port emulation entity.

3.1 SERVICE DEFINITION MODEL

The figure below shows a model of how RFCOMM fits into a typical system. This figure represents the RFCOMM reference model.

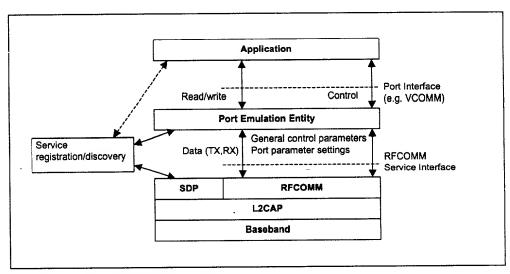


Figure 3.1: RFCOMM reference model

The elements of the RFCOMM reference model are described below.

Element	Description
Application	Applications that utilize a serial port communication interface
Port Emulation Entity	The port emulation entity maps a system-specific communication interface (API) to the RFCOMM services. The port emulation entity plus RFCOMM make up a port driver
RFCOMM	Provides a transparent data stream and control channel over an L2CAP channel. Multiplexes multiple emulated serial ports
Service Registra- tion/ Discovery	Server applications register here on local device, and it provides services for client applications to discover how to reach server applications on other devices
L2CAP	Protocol multiplexing, SAR
Baseband	Baseband protocols defined by BT

Bluetooth.

4 TS 07.10 SUBSET SUPPORTED BY RFCOMM

4.1 OPTIONS AND MODES

RFCOMM uses the basic option of TS 07.10.

4.2 FRAME TYPES

Table 4.1 shows the TS 7.10 frame types that are supported in RFCOMM.

Frame Types	
Set Asynchronous Balanced	Mode (SABM) command
Unnumbered Acknowledgem	ent (UA) response
Disconnected Mode (DM) res	sponse
Disconnect (DISC) command	
Unnumbered information with	n header check (UIH) command and response

Table 4.1: Supported frame types in RFCOMM

The 'Unnumbered Information (UI) command and response' are not supported by RFCOMM. Since the error recovery mode option of the TS 07.10 protocol is not used in RFCOMM none of the associated frame types are supported.

4.3 COMMANDS

TS 07.10 defines a multiplexer that has a dedicated control channel, DLCI 0. The control channel is used to convey information between two multiplexers. The following commands in TS 07.10 are supported by RFCOMM:

Supported Control Channel Commands	The second secon
Test Command (Test)	
Flow Control On Command (Fcon)	
Flow Control Off Command (Fcoff)	
Modern Status Command (MSC)	
Remote Port Negotiation Command (RPN)	
Remote Line Status (RLS)	
DLC parameter negotiation (PN)	
Non Supported Command Response (NSC)	

Whenever a non-supported command type is received a 'Non-Supported Command Response (NSC)' should be sent.

Bluetooth.

4.4 CONVERGENCE LAYERS

RFCOMM only supports the type 1 convergence layer in TS 07.10.

The Modem Status Command (MSC) shall be used to convey the RS-232 control signals and the break signal for all emulated serial ports.

Bluetooth.

5 TS 07.10 ADAPTATIONS FOR RFCOMM

5.1 MEDIA ADAPTATION

The opening flag and the closing flags in the 07.10 basic option frame are not used in RFCOMM, instead it is only the fields contained between the flags that are exchanged between the L2CAP layer and RFCOMM layer, see Figure 5.1.

Flag	Address	Control	Length Indicator	Information	FCS	Flag
0111 1101	1 octet	1 octet	1 or 2 octets	Unspecified length but integral number of octets	1 octet	0111 1101

Figure 5.1: Frame Structure for Basic option. Note that the opening and closing flags from the 07.10 Basic option are excluded in RFCOMM.

5.1.1 FCS calculation

In 07.10, the frame check sequence (FCS) is calculated on different sets of fields for different frame types. These are the fields that the FCS are calculated on:

For SABM, DISC, UA, DM frames: on Address, Control and length field.

For UIH frames: on Address and Control field.

(This is stated here for clarification, and to set the standard for RFCOMM; the fields included in FCS calculation have actually changed in version 7.0.0 of TS 07.10, but RFCOMM will not change the FCS calculation scheme from the one above.)

Bluetooth.

5.2 TS 07.10 MULTIPLEXER START-UP AND CLOSEDOWN PROCEDURE

The start-up and closedown procedures as specified in section 5.7 in TS 07.10 are not supported. This means that the AT-command AT+CMUX is not supported by RFCOMM, neither is the multiplexer close down (CLD) command.

At any time, there must be at most one RFCOMM session between any pair of devices. When establishing a new DLC, the initiating entity must check if there already exists an RFCOMM session with the remote device, and if so, establish the new DLC on that. A session is identified by the Bluetooth BD_ADDR of the two endpoints¹.

5.2.1 Start-up procedure

The device opening up the first emulated serial port connection between two devices is responsible for first establishing the multiplexer control channel. This involves the following steps:

- Establish an L2CAP channel to the peer RFCOMM entity, using L2CAP service primitives, see L2CAP "Service Primitives" on page 295.
- Start the RFCOMM multiplexer by sending SABM command on DLCI 0, and await UA response from peer entity. (Further optional negotiation steps are possible.)

After these steps, DLCs for user data traffic can be established.

5.2.2 Close-down procedure

The device closing the last connection (DLC) on a particular session is responsible for closing the multiplexer by closing the corresponding L2CAP channel.

Closing the multiplexer by first sending a DISC command frame on DLCI 0 is optional, but it is mandatory to respond correctly to a DISC (with UA response).

5.2.3 Link loss handling

If an L2CAP link loss notification is received, the local RFCOMM entity is responsible for sending a connection loss notification to the port emulation/proxy entity for each active DLC. Then all resources associated with the RFCOMM session should be freed.

The appropriate action to take in the port emulation/proxy entity depends on the API on top. For example, for an emulated serial port (vCOMM), it would be suitable to drop CD, DSR and CTS signals (assuming device is a DTE).

This implies that, when responding to an L2CAP connection indication, the RFCOMM entity should save and associate the new RFCOMM session with the remote BD_ADDR. This is, at least, necessary if subsequent establishment of a DLC in the opposite direction is possible (which may depend on device capabilities).

Bluetooth.

5.3 SYSTEM PARAMETERS

Table 5.1 contains all the applicable system parameters for the RFCOMM implementation of the TS 07.10 multiplexer.

System Parameter	Value
Maximum Frame Size (N1)	Default: 127 (negotiable range 23 – 32767)
Acknowledgement Timer (T1)	60 seconds
Response Timer for Multiplexer Control Channel (T2)	60 seconds

Table 5.1: System parameter values

Note: The timer T1 is the timeout for *frames* sent with the P/F-bit set to one (this applies only to SABM and DISC frames in RFCOMM). T2 is the timeout for *commands* sent in UIH frames on DLCI 0.

Since RFCOMM relies on lower layers to provide reliable transmission, the default action performed on timeouts is to close down the multiplexer session. The only exception to this is when trying to set up a new DLC on an existing session; i.e. waiting for the UA response for a SABM command. In this case, the initiating side may defer the timeout by an unspecified amount of time if it has knowledge that the delay is due to user interaction (e.g. authentication procedure in progress). When/if the connection attempt is eventually considered to have timed out, the initiating side must send a DISC command frame on the same DLCI as the original SABM command frame, in order to notify the other party that the connection attempt is aborted. (After that the initiating side will, as usual, expect a UA response for the DISC command.)

5.4 DLCI ALLOCATION WITH RFCOMM SERVER CHANNELS

To account for the fact that both client and server applications may reside on both sides of an RFCOMM session, with clients on either side making connections independent of each other, the DLCI value space is divided between the two communicating devices using the concept of RFCOMM server channels and a direction bit.

The RFCOMM server channel number is a subset of the bits in the DLCI part of the address field in the TS 07.10 frame.

Bit No.	1	2	3 4 5 6 7 8
TS 07.10	EA	C/R	DLCI
RFCOMM	EA	C/R	D Server Channel

Table 5.2: The format of the Address Field

Bluetooth.

Server applications registering with an RFCOMM service interface are assigned a Server Channel number in the range 1...30. [0 and 31 should not be used since the corresponding DLCIs are reserved in TS 07.10] It is this value that should be registered in the Service Discovery Database, see Section 7.2.

For an RFCOMM session, the initiating device is given the direction bit D=1 (and conversely, D=0 in the other device). When establishing a new data link connection on an existing RFCOMM session, the direction bit is used in conjunction with the Server Channel to determine the DLCI to use to connect to a specific application. This DLCI is thereafter used for all packets in both directions between the endpoints.

In effect, this partitions the DLCI value space such that server applications on the non-initiating device are reachable on DLCIs 2,4,6,...,60; and server applications on the initiating device are reachable on DLCIs 3,5,7,...,61. (Note that for a device that supports multiple simultaneous RFCOMM sessions to two or more devices, the direction bit might not be the same on all sessions.)

An RFCOMM entity making a new DLC on an existing session forms the DLCI by combining the Server Channel for the application on the other device, and the inverse of its own direction bit for the session.

DLCI 1 and 62-63 are reserved and never used in RFCOMM.

5.5 MULTIPLEXER CONTROL COMMANDS

Note that in TS 07.10, some Multiplexer Control commands pertaining to specific DLCIs may be exchanged on the control channel (DLCI 0) before the corresponding DLC has been established. (This refers the PN and RPN commands.) All such states associated with an individual DLC must be reset to their default values upon receiving a DISC command frame, or when closing the DLC from the local side. This is to ensure that all DLC (re-)establishments on the same session will have predictable results, irrespective of the session history.

5.5.1 Remote Port Negotiation Command (RPN)

The RPN command can be used before a new DLC is opened and should be used whenever the port settings change.

The RPN command is specified as optional in TS 07.10, but it is mandatory to recognize and respond to it in RFCOMM. (Although the handling of individual settings are implementation-dependent.)

Bluetooth.

5.5.2 Remote Line Status Command (RLS)

This command is used for indication of remote port line status.

The RLS command is specified as optional in TS 07.10, but it is mandatory to recognize and respond to it in RFCOMM. (Although the handling of individual settings are implementation-dependent.)

5.5.3 DLC parameter negotiation (PN)

The PN command is specified as optional in TS 07.10, but it is mandatory to recognize and respond to it in RFCOMM. This command can be used before a new DLC is opened.

There are some parameters in the PN command which convey information not applicable to RFCOMM. These fields must therefore be set to predetermined values by the sender, and they must be ignored by the receiver. This concern the following fields (see table 3 in ref. [1]):

- I1-I4 must be set to 0. (Meaning: use UIH frames.)
- CL1-CL4 must be set to 0. (Meaning: use convergence layer type 1.)
- T1-T8 must be set to 0. (Meaning: acknowledgment timer *T1*, which is not negotiable in RFCOMM.)
- NA1-NA8 must be set to 0. (Meaning: number of retransmissions N2; always 0 for RFCOMM)
- K1-K3 must be set to 0. (Meaning: defines the window size for error recovery mode, which is not used for RFCOMM.)

If a command is received with invalid (or for some reason unacceptable) values in any field, a DLC parameter negotiation response must be issued with values that are acceptable to the responding device.

Bluetooth.

6 FLOW CONTROL

Wired ports commonly use flow control such as RTS/CTS to control communications. On the other hand, the flow control between RFCOMM and the lower layer L2CAP depends on the service interface supported by the implementation. In addition RFCOMM has its own flow control mechanisms. This section describes the different flow control mechanisms.

6.1 L2CAP FLOW CONTROL IN OVERVIEW

L2CAP relies on the flow control mechanism provided by the Link Manager layer in the baseband. The flow control mechanism between the L2CAP and RFCOMM layers is implementation-specific.

6.2 WIRED SERIAL PORT FLOW CONTROL

Wired Serial ports falls into two camps – software flow control using characters such as XON/XOFF, and flow control using RTS/CTS or DTR/DSR circuits. These methods may be used by both sides of a wired link, or may be used only in one direction.

6.3 RFCOMM FLOW CONTROL

The RFCOMM protocol provides two flow control mechanisms:

- 1. The RFCOMM protocol contains flow control commands that operate on the aggregate data flow between two RFCOMM entities; i.e. all DLCIs are affected. The control channel commands, FCon and FCoff, are defined in section 5.4.6.3 in ref [1].
- 2. The Modern Status command as defined in section 5.4.6.3 in ref [1] is the flow control mechanism that operates on individual DLCI.

6.4 PORT EMULATION ENTITY SERIAL FLOW CONTROL

On Type 1 devices some port drivers (Port Emulation Entities plus RFCOMM) will need to provide flow control services as specified by the API they are emulating. An application may request a particular flow control mechanism like XON/XOFF or RTS/CTS and expect the port driver to handle the flow control. On type 2 devices the port driver may need to perform flow control on the non-RFCOMM part of the communication path; i.e. the physical RS-232 port. This flow control is specified via the control parameters sent by the peer RFCOMM entity (usually a type 1 device). The description of flow control in this section is for port drivers on type 1 devices.

Since RFCOMM already has its own flow control mechanism, the port driver does not need to perform flow control using the methods requested by the application. In the ideal case, the application sets a flow control mechanism

Bluetooth.

and assumes that the COMM system will handle the details. The port driver could then simply ignore the request and rely on RFCOMM's flow control. The application is able to send and receive data, and does not know or care that the port driver did not perform flow control using the mechanism requested. However, in the real world some problems arise.

- The RFCOMM-based port driver is running on top of a packet-based protocol where data may be buffered somewhere in the communication path. Thus, the port driver cannot perform flow control with the same precision as in the wired case.
- The application may decide to apply the flow control mechanism itself in addition to requesting flow control from the port driver.

These problems suggest that the port driver must do some additional work to perform flow control emulation properly. Here are the basic rules for flow control emulation.

- The port driver will not solely rely on the mechanism requested by the application but use a combination of flow control mechanisms.
- The port driver must be aware of the flow control mechanisms requested by the application and behave like the wired case when it sees changes on the non-data circuits (hardware flow control) or flow control characters in the incoming data (software flow control). For example, if XOFF and XON characters would have been stripped in the wired case they must be stripped by the RFCOMM based port driver.
- If the application sets a flow control mechanism via the port driver interface
 and then proceeds to invoke the mechanism on its own, the port driver must
 behave in a manner similar to that of the wired case (e.g. If XOFF and XON
 characters would have been passed through to the wire in the wired case
 the port driver must also pass these characters).

These basic rules are applied to emulate each of the wired flow control schemes. Note that multiple types of flow control can be set at the same time. Section 5.4.8 in ref [1] defines each flow control mechanism.

Bluetooth.

7 INTERACTION WITH OTHER ENTITIES

7.1 PORT EMULATION AND PORT PROXY ENTITIES

This section defines how the RFCOMM protocol should be used to emulate serial ports. Figure 7.1 shows the two device types that the RFCOMM protocol supports.

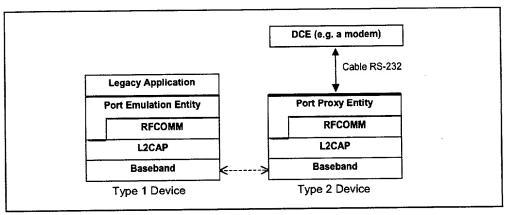


Figure 7.1: The RFCOMM communication model

Type 1 devices are communication endpoints such as computers and printers. Type 2 devices are part of a communication segment; e.g. modems.

7.1.1 Port Emulation Entity

The port emulation entity maps a system specific communication interface (API) to the RFCOMM services.

7.1.2 Port Proxy Entity

The port proxy entity relays data from RFCOMM to an external RS-232 interface linked to a DCE. The communications parameters of the RS-232 interface are set according to received RPN commands, see Section 5.5.1.

7.2 SERVICE REGISTRATION AND DISCOVERY

Registration of individual applications or services, along with the information needed to reach those (i.e. the RFCOMM Server Channel) is the responsibility of each application respectively (or possibly a Bluetooth configuration application acting on behalf of legacy applications not directly aware of Bluetooth).

Below is a template/example for developing service records for a given service or profile using RFCOMM. It illustrates the inclusion of the ServiceClassList with a single service class, a ProtocolDescriptor List with two protocols

Bluetooth.

(although there may be more protocols on top of RFCOMM). The example shows the use of one other universal attribute (ServiceName). For each service running on top of RFCOMM, appropriate SDP-defined universal attributes and/or service-specific attributes will apply. For additional information on Service Records, see the SDP Specification, Section 2.2 on page 332.

The attributes that a client application needs (at a minimum) to connect to a service on top of RFCOMM are the ServiceClassIDList and the ProtocolDescriptorList (corresponding to the shaded rows in the table below).

ltem	Definition	Type/Size	Value	Attribute ID
ServiceClassIDList			Note1	0x0001
ServiceClass0	Note5	UUIÐ/32-bit	Note1	
ProtocolDescriptorList				0x0004
Protocol0	L2CAP	UUID/32-bit	L2CAP /Note1	
Protocoi1	RFCOMM	UUID/32-bit	RFCOMM /Note1	
ProtocolSpecificParameter0	Server Channel	Uint8	N = server channel #	
[other protocols]		UUID/32-bit	Note1	
[other protocol-specific parameters]	Note3	Note3	Note3	
ServiceName	Displayable text name	DataElement/ String	'Example service'	Note2
[other universal attributes as appropriate for this service]	Note4	Note4	Note4	Note4
[service-specific attributes]	Note3	Note3	Note3	Note3

Notes:

- 1. Defined in "Bluetooth Assigned Numbers" on page 1009.
- 2. For national language support for all 'displayable' text string attributes, an offset has to be added to the LanguageBaseAttributeIDList value for the selected language (see the SDP Specification, Section 5.1.14 on page 365 for details).
- 3. To be defined (where necessary) for the specific service.
- 4. For a specific service some of the SDP-defined universal attributes may apply. See the SDP Specification, Section 5.1 on page 358.
- 5. This indicates the class of service. It can be a single entry or a list of service classes ranging from generic to most specific.

Bluetooth.

7.3 LOWER LAYER DEPENDENCIES

7.3.1 Reliability

RFCOMM uses the services of L2CAP to establish L2CAP channels to RFCOMM entities on other devices. An L2CAP channel is used for the RFCOMM/TS 07.10 multiplexer session. On such a channel, the TS 07.10 frames listed in Section 4.2 are sent, with the adaptation defined in Section 5.1.

Some frame types (SABM and DISC) as well as UIH frames with multiplexer control commands sent on DLCI 0 always require a response from the remote entity, so they are acknowledged on the RFCOMM level (but not retransmitted in the absence of acknowledgment, see Section 5.3). Data frames do not require any response in the RFCOMM protocol, and are thus unacknowledged.

Therefore, RFCOMM must require L2CAP to provide channels with maximum reliability, to ensure that all frames are delivered in order, and without duplicates. Should an L2CAP channel fail to provide this, RFCOMM expects a link loss notification, which should be handled by RFCOMM as described in Section 5.2.3.

7.3.2 Low power modes

If all L2CAP channels towards a certain device are idle for a certain amount of time, a decision may be made to put that device in a low power mode (i.e. use hold, sniff or park, see 'Baseband Specification' Section 10.10.3 on page 125). This will be done without any interference from RFCOMM. RFCOMM can state its latency requirements to L2CAP. This information may be used by lower layers to decide which low power mode(s) to use.

The RFCOMM protocol as such does not suffer from latency delays incurred by low power modes, and consequentially, this specification does not state any maximum latency requirement on RFCOMM's behalf. Latency sensitivity inherently depends on application requirements, which suggests that an RFCOMM service interface implementation could include a way for applications to state latency requirements, to be aggregated and conveyed to L2CAP by the RFCOMM implementation. (That is if such procedures make sense for a particular platform.)

Bluetooth.

8 REFERENCES

- [1] TS 07.10, ver 6.3.0, ETSI
- [2] Bluetooth L2CAP Specification
- [3] Bluetooth SDP Specification
- [4] Bluetooth Assigned Numbers

Bluetooth.

9 TERMS AND ABBREVIATIONS

The following terms are used throughout the document.

DTE Data Terminal Equipment – in serial communications, DTE refers to a

device at the endpoint of the communications path; typically a com-

puter or terminal

DCE Data Circuit-Terminating Equipment – in serial communications, DCE

refers to a device between the communication endpoints whose sole task is to facilitate the communications process; typically a modem

RFCOMM initiator The device initiating the RFCOMM session; i.e.setting up RFCOMM

channel on L2CAP and starting RFCOMM multiplexing with the SABM

command frame on DLCI 0 (zero)

RFCOMM Client An RFCOMM client is an application that requests a connection to

another application (RFCOMM server)

RFCOMM Server An RFCOMM server is an application that awaits a connection from an

RFCOMM client on another device. What happens after such a con-

nection is established is not within the scope of this definition

RFCOMM Server

Channel

This is a subfield of the TS 07.10 DLCI number. This abstraction is used to allow both server and client applications to reside on both sides

of an RFCOMM session

Bluetooth.

Part F:2

IrDA INTEROPERABILITY

The IrOBEX protocol is utilized by the Bluetooth technology. In Bluetooth, OBEX offers the same features for applications as within the IrDA protocol hierarchy, enabling the applications to work over the Bluetooth protocol stack as well as the IrDA stack.

How the true that the true

í,ħ

Bluetooth.

Bluetooth.

CONTENTS

1	Intro	duction	414		
•	1.1	OBEX and Bluetooth Architecture	415		
	1.2	Bluetooth OBEX-Related Specifications			
	1.3	Other IrOBEX Implementations	416		
2		X Object and Protocol			
_	2.1	Object	417		
	2.2	Session Protocol			
		2.2.1 Connect Operation	418		
		2.2.2 Disconnect Operation			
		2.2.3 Put Operation			
		2.2.4 Get Operation			
		2.2.5 Other Operations			
3	ORE				
	3.1	X over RFCOMMOBEX Server Start-up on RFCOMM	421		
	3.2	Receiving OBEX Packets from Serial Port	421		
	3.3	Connection Establishment	422		
	3.4	Disconnection	422		
	3.5	Pushing and Pulling OBEX Packets over RFCOMM	422		
4		EX over TCP/IP			
**	4.1	OBEX Server Start-up on TCP/IP	423		
	4.2	Connection Establishment	423		
	4.3	Disconnection	424		
	4.4	Pushing and Pulling OBEX Packets over TCP			
5					
J	5.1	etooth Application Profiles using OBEX	425		
	5.2	File Transfer	425		
	5.3	Object Push			
6	-	erences			
7		of Acronyms and Abbreviations			
7	LIST	. OI ACIUIVIIID CIIU MUUICYICIIVIID			

Bluetooth.

1 INTRODUCTION

The goal of this document is to enable the development of application programs that function well over both short-range RF and IR media. Each media type has its advantages and disadvantages but the goal is for applications to work over both. Rather than fragment the application domain, this document defines the intersection point where Bluetooth and IrDA applications may converge. That intersection point is IrOBEX [1].

IrOBEX is a session protocol defined by IrDA. This protocol is now also utilized by the Bluetooth technology, making it possible for applications to use either the Bluetooth radio technology or the IrDA IR technology. However, even though both IrDA and Bluetooth are designed for short-range wireless communications, they have some fundamental differences relating to the lower-layer protocols. IrOBEX will therefore be mapped over the lower layer protocols which are adopted by Bluetooth.

This document defines how IrOBEX (OBEX for short) is mapped over RFCOMM [2] and TCP/IP [3]. Originally, OBEX (Object Exchange Protocol) was developed to exchange data objects over an infrared link and was placed within the IrDA protocol hierarchy. However, it can appear above other transport layers, now RFCOMM and TCP/IP. At this moment, it is worth mentioning that the OBEX over TCP/IP implementation is an optional feature for Bluetooth devices supporting the OBEX protocol.

The IrOBEX specification [1] provides a model for representing objects and a session protocol, which structures the dialogue between two devices. The IrOBEX protocol follows a client/server **request-response** paradigm for the conversation format.

Bluetooth uses only the connection-oriented OBEX even though IrDA has specified the connectionless OBEX also. The reasons for the connection-oriented approach are:

- OBEX is mapped over the connection-oriented protocols in the Bluetooth architecture.
- Most of application profiles using OBEX and Bluetooth needs a connectionoriented OBEX to provide the functionality described for the features included in these profiles.
- The connectionless OBEX with the connection-oriented one would raise the interoperability problems, which are not desirable.

Bluetooth.

1.1 OBEX AND BLUETOOTH ARCHITECTURE

Figure 1.1 depicts part of the hierarchy of the Bluetooth architecture and shows the placement of the OBEX protocol and the application profiles using it (See also Section 5 on page 425). The protocols can also communicate with the service discovery DB even though the figure does not show it.

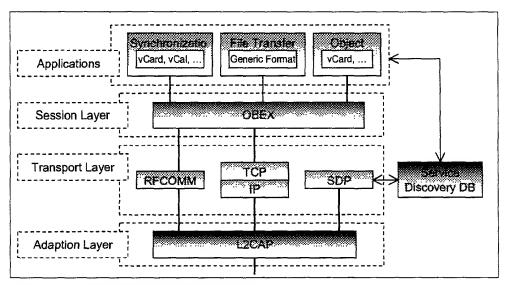


Figure 1.1: Part of Bluetooth Protocol Hierarchy

In the Bluetooth system, the purpose of the OBEX protocol is to enable the exchange of data objects. The typical example could be an object push of business cards to someone else. A more complex example is synchronizing calendars on multiple devices using OBEX. Also, the File Transfer applications can be implemented using OBEX. For the Object Push and Synchronization applications, content formats can be the vCard [4], vCalendar [5], vMessage [6], and vNotes [6] formats. The vCard, vCalendar, vMessage, and vNotes describe the formats for the electronic business card, the electronic calendaring and scheduling, the electronic message and mails, and the electronic notes, respectively.

1.2 BLUETOOTH OBEX-RELATED SPECIFICATIONS

Bluetooth Specification includes five separate specifications related to OBEX and applications using it:

- 1. Bluetooth IrDA Interoperability Specification (This specification)
- Defines how the applications can function over both Bluetooth and IrDA
- Specifies how OBEX is mapped over RFCOMM and TCP
- · Defines the application profiles using OBEX over Bluetooth

Bluetooth.

- 2. Bluetooth Generic Object Exchange Profile Specification [7]
- Generic interoperability specification for the application profiles using OBEX
- Defines the interoperability requirements of the lower protocol layers (e.g. Baseband and LMP) for the application profiles
- 3. Bluetooth Synchronization Profile Specification [8]
- Application Profile for the Synchronization applications
- Defines the interoperability requirements for the applications within the Synchronization application profile
- Does <u>not</u> define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.
- 4. Bluetooth File Transfer Profile Specification [9]
- · Application Profile for the File Transfer applications
- Defines the interoperability requirements for the applications within the File Transfer application profile.
- Does <u>not</u> define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.
- 5. Bluetooth Object Push Profile Specification [10]
- · Application Profile for the Object Push applications
- Defines the interoperability requirements for the applications within the Object Push application profile.
- Does <u>not</u> define the requirements for the Baseband, LMP, L2CAP, or RFCOMM.

1.3 OTHER IROBEX IMPLEMENTATIONS

Over IR, OBEX has also been implemented over IrCOMM and Tiny TP. The Bluetooth technology does not define these protocols as transport protocols for OBEX, but they can be supported by independent software vendors if desired.

Bluetooth.

2 OBEX OBJECT AND PROTOCOL

This section is dedicated to the model of OBEX objects and the OBEX session protocol. The section is intended to be read with the IrOBEX specification [1].

2.1 OBJECT

The OBEX object model (Section 2 in [1]) describes how OBEX objects are presented. The OBEX protocol can transfer an object by using the **Put**- and **Get**-operations (See Section 2.2.3 and 2.2.4). One object can be exchanged in one or more **Put**-requests or **Get**-responses.

The model handles both information about the object (e.g. type) and object itself. It is composed of headers, which consist of a header ID and value (See Section 2.1 in [1]). The header ID describes what the header contains and how it is formatted, and the header value consists of one or more bytes in the format and meaning specified by Header ID. The specified headers are Count, Name, Type, Length, Time, Description, Target, HTTP, Body, End of Body, Who, Connection ID, Application Parameters, Authenticate Challenge, Authenticate Response, Object Class, and User-Defined Headers. These are explained in detail by Section 2.2 in the IrOBEX specification.

2.2 SESSION PROTOCOL

The OBEX operations are formed by **response-request** pairs. Requests are issued by the client and responses by the server. After sending a request, the client waits for a response from the server before issuing a new request. Each request packet consists of a one-byte opcode (See Section 3.3 in [1]), a two-byte length indicator, and required or optional data depending on the operation. Each response packet consists of a one-byte response code (See Section 3.2.1 in [1]), a two-byte length indicator, and required or optional data depending on the operation.

In the following subsections, the OBEX operations are explained in general.

Bluetooth.

2.2.1 Connect Operation

An OBEX session is started, when an application asks the first time to transmit an OBEX object. An OBEX client starts the establishment of an OBEX connection. The session is started by sending a **Connect**-request (See Section 3.3.1 in [1]). The request format is:

Byte 0	Bytes 1 and 2	Byte 3	Byte 4	Bytes 5 and 6	Byte 7 to n
0x80 (opcode)	Connect request packet length	OBEX version number	Flags	Maximum OBEX packet length	Optional headers

Note. The Big Endian format is used to define the byte ordering for the PDUs (requests and responses) in this specification as well as in the IrOBEX specification; i.e. the most significant byte (MSB) is always on left and the least significant byte (LSB) on right.

At the remote host, the **Connect**-request is received by the OBEX server, if it exists. The server accepts the connection by sending the successful response to the client. Sending any other response (i.e. a non-successful response) back to the client indicates a failure to make a connection. The response format is:

Byte 0	Bytes 1 and 2	Byte 3	Byte 4	Bytes 5 and 6	Byte 7 to n
Response code	Connect response packet length	OBEX version number	Flags	Maximum OBEX packet length	Optional headers

The response codes are list in the Section 3.2.1 in the IrOBEX specification. The bytes 5 and 6 define the maximum OBEX packet length, which can be received by the server. This value may differ from the length, which can be received by the client. These **Connect**-request and response packets must each fit in a single packet.

Once a connection is established it remains 'alive', and is only disconnected by requests/responses or by failures (i.e. the connection is not automatically disconnected after each OBEX object has completely transmitted).

Bluetooth.

2.2.2 Disconnect Operation

The disconnection of an OBEX session occurs when an application, which is needed for an OBEX connection, is closed or the application wants to change the host to which the requests are issued. The client issues the **Disconnect**-request (See Section 3.3.2 in [1]) to the server. The request format is:

Byte 0	Bytes 1 and 2	Byte 3
0x81	Packet length	Optional headers

The request cannot be refused by the server. Thus, it has to send the response, and the response format is:

Byte 0	Bytes 1 and 2	Byte 3
0xA0	Response packet length	Optional response headers

2.2.3 Put Operation

When the connection has been established between the client and server the client is able to push OBEX objects to the server. The **Put**-request is used to push an OBEX object (See Section 3.3.3 in [1]). The request has the following format.

Byte 0	Bytes 1 and 2	Byte 3
0x02 (0x82 when Final bit set)	Packet length	Sequence of headers

A **Put**-request consists of one or more request packets, depending on how large the transferred object is, and how large the packet size is. A response packet from the server is required for every **Put**-request packet. Thus, one response is not permitted for several request packets, although they consist of one OBEX object. The response format is:

Byte 0	Bytes 1 and 2	Byte 3
Response code	Response packet length	Optional response headers

Bluetooth.

2.2.4 Get Operation

When the connection has been established between the client and server, the client is also able to pull OBEX objects from the server. The **Get**-request is used to pull an OBEX object (See Section 3.3.4 in [1]). The request has the following format.

Byte 0	Bytes 1 and 2	Byte 3
0x03 (0x83 when Final bit set)	Packet length	Sequence of headers starting with Name

The object is returned as a sequence of headers, and the client has to send a request packet for every response packet. The response format is:

Byte 0	Bytes 1 and 2	Byte 3
Response code	Response packet length	Optional response headers

2.2.5 Other Operations

Other OBEX operations consist of a **SetPath**-, and an **Abort**-operation. These are carefully explained in the Sections 3.3.5-6 in the IrOBEX specification. It is important to note that the client can send an **Abort**-request after each response – even in the middle of a request/response sequence. Thus, the whole OBEX object does <u>not</u> have to be received before sending an **Abort**-request. In addition to these operations, the IrOBEX specification facilitates user-defined operations, but their use may not necessarily be adopted in Bluetooth.

Bluetooth.

3 OBEX OVER RFCOMM

This section specifies how OBEX is mapped over RFCOMM, which is the multiplexing and transport protocol based on ETSI TS 07.10 [11] and which also provides a support for serial cable emulation. The Bluetooth devices supporting the OBEX protocol must satisfy the following requirements.

- The device supporting OBEX must be able to function as either a client, a server, or both
- 2. All servers running simultaneously on a device must use separate RFCOMM server channels
- 3. Applications (service/server) using OBEX must be able to register the proper information into the service discovery database. This information for different application profiles is specified in the profile specifications

3.1 OBEX SERVER START-UP ON RFCOMM

When a client sends a connecting request, a server is assumed to be ready to receive requests. However, before the server is ready to receive (i.e. is running) certain prerequisites must be fulfilled before the server can enter the listening mode:

- 1. The server must open an RFCOMM server channel
- 2. The server must register its capabilities into the service discovery database

After this, other hosts are able to find the server if needed, and the server listens for get requests from clients.

3.2 RECEIVING OBEX PACKETS FROM SERIAL PORT

As discussed earlier, one object can be exchanged over one or more **Put**-requests or **Get**-responses (i.e. the object is received in one or several packets). However, if OBEX is running directly over the serial port, it does not receive packets from RFCOMM. Instead, a byte stream is received by OBEX from a serial port emulated by RFCOMM.

To detect packets in the byte stream, OBEX has to look for opcodes or response codes (See Chapter 2.2) depending on whether a packet is a request or a response. The opcodes and response code can be thought of as the start flags of packets. In OBEX packets, there is no 'end flag' that would indicate the end of a packet. However, after the opcode or response code, the length of a packet is received in the next two bytes. Thus, the whole length of a packet is known, and the boundary of two packets can be determined.

Bluetooth.

All data that is not recognized must be dumped. This could cause a synchronization problem but, considering the nature of the OBEX protocol, this is not a problem over RFCOMM, which provides reliable transport over Bluetooth.

3.3 CONNECTION ESTABLISHMENT

A client initiates the establishment of a connection. However, the following sequence of tasks must occur before the client is able to send the first request for data:

- By using the SD protocol described in the SDP specification [12], the client must discover the proper information (e.g. RFCOMM channel) associated with the server on which the connection can be established
- 2. The client uses the discovered RFCOMM channel to establish the RFCOMM connection
- The client sends the Connect-request to the server, to establish an OBEX session. The session is established correctly if the client receives a successful response from the server

3.4 DISCONNECTION

The disconnection of an OBEX session over RFCOMM is straightforward. The disconnection is done by using the **Disconnect**-request (See Section 2.2.2). When the client has received the response, the next operation is to close the RFCOMM channel assigned to the OBEX client.

3.5 PUSHING AND PULLING OBEX PACKETS OVER RFCOMM

Data is pushed in OBEX packets over RFCOMM by using **Put**-requests (See Section 2.2.3). After each request, a response is required before the next request with the data can be pushed.

Pulling data from a remote host happens by sending a **Get**-request (See Section 2.2.4. The data arrives in OBEX response packets. After each response, a new request has to be sent, to pull more data.

Bluetooth.

4 OBEX OVER TCP/IP

This section specifies how OBEX is mapped over the TCP/IP creating reliable connection-oriented services for OBEX. This specification does <u>not</u> define how TCP/IP is mapped over Bluetooth.

The Bluetooth devices, which support the OBEX protocol over TCP/IP, must satisfy the following requirements:

- 1. The device supporting OBEX must be able to function as either a client, or a server, or both
- For the server, the TCP port number 650 is assigned by IANA. If an assigned number is not desirable, the port number can be a value above 1023. However, the use of the TCP port number (650) defined by IANA is highly recommended. The 0-1023 range is reserved by IANA (See [13])
- 3. The client must use a port number (on the client side), which is not within the 0-1023 range
- 4. Applications (service/server) using OBEX must be able to register the proper information into the service discovery database. This information for different application profiles is specified in the profile specifications

4.1 OBEX SERVER START-UP ON TCP/IP

When a client sends a **Put-** or **Get-**request, a server is assumed to be ready to receive requests. However, when the server is ready (i.e. is running), certain prerequisites must be fulfilled before the server can enter the listening mode:

- 1. The server must initialize a TCP port with the value 650 or value above 1023
- 2. The server registers its capabilities into the service discovery database

After this, other devices are able to find the server if needed, and the server listens for get requests from clients.

4.2 CONNECTION ESTABLISHMENT

A client initiates a connection. However, the following sequence of tasks must occur before a connection can be established:

- 1. By using, the SD protocol described in the SDP specification [12], the client discovers the proper information (e.g. TCP port number) associated with the server, to enable the connection can be established
- 2. The client initializes a socket associated to a TCP port number above 1023, and establishes a TCP connection with the host of the server
- 3. The client sends the **Connect**-request to the server, to establish an OBEX session. The session is established correctly if the client receives a successful response from the server.

Bluetooth.

4.3 DISCONNECTION

The disconnection of an OBEX session over TCP is straightforward. The disconnection is done by using the **Disconnect**-request (See Section 2.2.2). When the client has received the response, the next operation is to close the TCP port dedicated for this session.

4.4 PUSHING AND PULLING OBEX PACKETS OVER TCP

See Section 3.5.

Bluetooth.

5 BLUETOOTH APPLICATION PROFILES USING OBEX

Bluetooth SIG (Special Interest Group) has defined three separate application profiles using OBEX. These profiles are briefly introduced in this section.

5.1 SYNCHRONIZATION

Basically, the synchronization means comparing two object stores, determining their inequalities, and then unifying these two object stores. The Bluetooth devices supporting the synchronization may be desktop PCs, notebooks, PDAs, cellular phones, or smart phones.

The Bluetooth Synchronization profile uses the servers and clients compliant to the IrMC synchronization specified by IrDA (See Section 5 in [6]). The Bluetooth Synchronization servers and clients must support the level 4 synchronization functionality specified in the IrMC specification.

The actual logic of the synchronization engines which process the synchronization algorithm at the client device is implementation-specific. It is therefore left to the participating software vendors, and is not considered in the Bluetooth specifications.

The synchronization is not limited to one type of application. The Bluetooth synchronization (i.e. the IrMC synchronization) enables four different application classes:

- 1. Phone Book provides a means for a user to manage contact records
- 2. Calendar enables a user to manage calendar items, and can also be used for 'to-do' or task lists
- 3. Messaging lets a user manage messages (e.g. e-mails)
- 4. Notes provides a means for a user to manage small notes

The interoperability requirements for the Bluetooth Synchronization profile are defined in the Synchronization Profile [8] and Generic Object Exchange Profile [7] specifications.

5.2 FILE TRANSFER

At the minimum, the File Transfer profile is intended for sending and retrieving generic files to and from the Bluetooth device. The File Transfer service also facilitates the browsing of the remote Bluetooth device's folder.

The interoperability requirements for the Bluetooth File Transfer profile are defined in the File Transfer Profile [9] and Generic Object Exchange Profile [7] specifications.

-! Bluetooth.

5.3 OBJECT PUSH

The Object Push profile is the special case of the File Transfer Profile for beaming objects and optionally pulling the default objects. At a minimum, it offers the capability to exchange business cards, but is not limited to this service.

The interoperability requirements for the Object Push profile are defined in the Object Push Profile [10] and Generic Object Exchange Profile [7] specifications.

Bluetooth.

6 REFERENCES

- [1] Infrared Data Association, IrDA Object Exchange Protocol (IrOBEX), Version 1.2, April 1999
- [2] Bluetooth RFCOMM with TS 07.10, on page 385
- [3] Internet Engineering Task Force, IETF Directory List of RFCs (http://www.ietf.org/rfc/), May 1999.
- [4] The Internet Mail Consortium, vCard The Electronic Business Card Exchange Format, Version 2.1, September 1996.
- [5] The Internet Mail Consortium, vCalendar The Electronic Calendaring and Scheduling Exchange Format, Version 1.0, September 1996.
- [6] Infrared Data Association, IrMC (Ir Mobile Communications) Specification, Version 1.1, February 1999.
- [7] Bluetooth Generic Object Exchange Profile, see Volume 2.
- [8] Bluetooth Synchronization Profile, see Volume 2.
- [9] Bluetooth File Transfer Profile, see Volume 2.
- [10] Bluetooth Object Push Profile, see Volume 2.
- [11] ETSI, TS 07.10, Version 6.3.0
- [12] Bluetooth Service Discovery Protocol, see Volume 2.
- [13] Internet Assigned Numbers Authority, IANA Protocol/Number Assignments Directory (http://www.iana.org/numbers.html), May 1999.

Bluetooth.

7 LIST OF ACRONYMS AND ABBREVIATIONS

Abbreviation or Acronym	Meaning
GEOP	Generic Object Exchange Profile
IrDA	Infrared Data Association
IrMC	Ir Mobile Communications
L2CAP	Logical Link Control and Adaptation Protocol
LSB	Least Significant Byte
MSB	Most Significant Byte
OBEX	Object exchange protocol
PDU	Protocol Data Unit
RFCOMM	Serial cable emulation protocol based on ETSI TS 07.10
SD	Service Discovery
SDP	Service Discovery Protocol
SDDB	Service Discovery Database
TCP/IP	Transport Control Protocol/Internet Protocol

Part F:3

TELEPHONY CONTROL PROTOCOL SPECIFICATION

TCS Binary

The same same and the same

This document describes the Bluetooth
Telephony Control protocol Specification –
Binary (TCS Binary), using a bit-oriented
protocol. This protocol defines the call control
signalling for the establishment of speech and
data calls between Bluetooth devices.
In addition, it defines mobility management
procedures for handling Bluetooth TCS
devices.

Bluetooth.

Bluetooth.

CONTENTS

Gene	eral Des	cription	435
1.1	Overvi	ew	435
1.2	Operat	ion between devices	435
1.3			
Call	Control	(CC)	439
2.1		•	
2.2			
	2.2.1	Call Request	
	2.2.2	Bearer selection	440
	2.2.3	Overlap Sending	441
	2.2.4	Call Proceeding	441
		2.2.4.1 Call proceeding, enbloc sending	441
		2.2.4.2 Call proceeding, overlap sending	
	2.2.5	Call Confirmation	442
	2.2.6	Call Connection	442
	2.2.7	Call Information	443
	2.2.8	Non-selected user clearing	443
	2.2.9	In-band tones and announcements	443
	2.2.10	Failure of call establishment	444
	2.2.11	Call Establishment Message Flow	445
2.3	Call Cl	earing	446
	2.3.1	Normal Call Clearing	
	2.3.2	Abnormal Call Clearing	447
	2.3.3	Clear Collision	
	2.3.4	Call Clearing Message Flow	448
	1.1 1.2 1.3 Call 2.1 2.2	1.1 Overvir 1.2 Operat 1.3 Operat 1.3 Operat 1.3 Operat 1.3 Operat 1.3 Operat 1.3 Call St. 2.2.1 2.2.2 2.2.3 2.2.4 2.2.5 2.2.6 2.2.7 2.2.8 2.2.9 2.2.10 2.2.11 2.3 Call Cl. 2.3.1 2.3.2 2.3.3	1.2 Operation between devices. 1.3 Operation between layers. Call Control (CC). 2.1 Call States. 2.2 Call Establishment. 2.2.1 Call Request. 2.2.2 Bearer selection. 2.2.3 Overlap Sending. 2.2.4 Call Proceeding. 2.2.4.1 Call proceeding, enbloc sending. 2.2.4.2 Call proceeding, overlap sending. 2.2.4.3 Expiry of timer T310. 2.2.5 Call Confirmation. 2.2.6 Call Connection. 2.2.7 Call Information. 2.2.8 Non-selected user clearing. 2.2.9 In-band tones and announcements. 2.2.10 Failure of call establishment. 2.2.11 Call Establishment Message Flow. 2.3 Call Clearing. 2.3.1 Normal Call Clearing. 2.3.2 Abnormal Call Clearing.

Telep	phony Co	ntrol Prote	ocol Specification	Bluetooth.		
3	Gro	Group Management (GM)				
	3.1		iew			
	3.2		/ireless User Group	449		
		3.2.1	Description	449		
		3.2.2	Encryption within the WUG	450		
		3.2.3	Unconscious pairing	450		
	3.3	Obtain 3.3.1	Access RightsProcedure description			
		3.3.2	Message flow	451		
	3.4	Config 3.4.1	uration Distribution Procedure Description			
		3.4.2	Message flow	452		
	3.5	Fast ir 3.5.1	nter-member AccessListen Request			
		3.5.2	Listen Accept	453		
		3.5.3	Listen Reject by the WUG Master	454		
		3.5.4	Listen Reject by the WUG Member	454		
		3.5.5	Message flow	454		
4	Con	nection	less TCS (CL)	455		
5	Sup	Supplementary Services (SS)				
	5.1		456			
	5.2	DTMF 5,2,1	start & stopStart DTMF request			
		5.2.2	Start DTMF response			
		5.2.3	Stop DTMF request			
		5.2.4	Stop DTMF response			
		5.2.5	Message flow			
	5 2		•			
	5.3	Regist	er Recall	458		

Telep	hony Co	ntrol Proto	col Specification	Bluetooth.
6	Mes	sage for	mats	459
	6.1	Call Co 6.1.1	ntrol Message Formats	
		6.1.2	CALL PROCEEDING	460
		6.1.3	CONNECT	461
	_	6.1.4	CONNECT ACKNOWLEDGE	461
		6.1.5	DISCONNECT	462
		6.1.6	INFORMATION	462
		6.1.7	PROGRESS	463
		6.1.8	RELEASE	463
		6.1.9	RELEASE COMPLETE	464
		6.1.10	SETUP	464
		6.1.11	SETUP ACKNOWLEDGE	465
		6.1.12	Start DTMF	465
		6.1.13	Start DTMF Acknowledge	466
		6.1.14	Start DTMF Reject	466
		6.1.15	Stop DTMF	466
		6.1.16	Stop DTMF Acknowledge	467
	6.2	Group 6.2.1	Management Message Formats ACCESS RIGHTS REQUEST	
		6.2.2	ACCESS RIGHTS ACCEPT	467
		6.2.3	ACCESS RIGHTS REJECT	468
		6.2.4	INFO SUGGEST	468
		6.2.5	INFO ACCEPT	468
		6.2.6	LISTEN REQUEST	469
		6.2.7	LISTEN SUGGEST	469
		6.2.8	LISTEN ACCEPT	469
		6.2.9	LISTEN REJECT	470
	6.3		onnectionless Message Formats	
		6.3.1	CL INFO	
7		_	ding	
	7.1 7.2		ewol Discriminator	
	7.2		ge Typege Type	
	7.4		nformation Elements	
		7.4.1	Coding rules	
		7.4.2	Audio control	
		7.4.3	Bearer capability	476
		7.4.4	Call class	479

Telepho	ony Con	trol Proto	col Specification	Bluetooth.
		7.4.5	Called party number	480
		7.4.6	Calling party number	481
		7.4.7	Cause	482
		7.4.8	Clock offset	482
		7.4.9	Company specific	483
		7.4.10	Configuration data	484
		7.4.11	Destination CID	485
		7.4.12	Keypad facility	485
		7.4.13	Progress indicator	485
		7.4.14	SCO Handle	486
		7.4.15	Sending complete	486
		7.4.16	Signal	486
8	Mess	age Err	or handling	487
	8.1	Protoco	ol Discrimination Error	487
	8.2	Messa	ge Too Short or Unrecognized	487
	8.3	Messa	ge Type or Message Sequence Errors	487
	8.4	Informa	ation Element Errors	487
9	Proto	ocol Par	ameters	489
	9.1	Protoc	ol Timers	489
10	Refe	rences		490
11	List	of Figure	es	491
12	List	of Table	s	492
Appe	ndix 1	- TCS (Call States	493

Bluetooth.

1 GENERAL DESCRIPTION

1.1 OVERVIEW

The Bluetooth Telephony Control protocol Specification Binary (TCS *Binary*) is based on the ITU-T Recommendation Q.931[1], applying the symmetrical provisions as stated in Annex D of Q.931. The resulting text does not discriminate between user and network side, but merely between Outgoing Side (the party originating the call) and Incoming Side (the party terminating the call). Effort was made to only apply those changes necessary for Bluetooth and foreseen applications, enabling re-use of Q.931 to the largest extent possible.

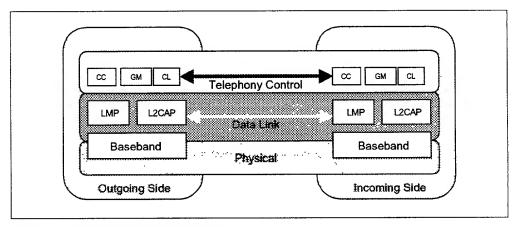


Figure 1.1: TCS within the Bluetooth stack

The TCS contains the following functionality:

- Call Control (CC) signalling for the establishment and release of speech and data calls between Bluetooth devices
- Group Management signalling to ease the handling of groups of Bluetooth devices
- ConnectionLess TCS (CL) provisions to exchange signalling information not related to an ongoing call

1.2 OPERATION BETWEEN DEVICES

TCS uses point-to-point signalling and may use point-to-multipoint signalling. Point-to-point signalling is used when it is known to which side (Bluetooth device) a call needs to be established (*single-point configuration*).

Point-to-multipoint signalling may be used when there are more sides available for call establishment (*multi-point configuration*); e.g. when, for an incoming call, a home base station needs to alert all phones in range.

Bluetooth.

Point-to-point signalling is mapped towards a connection-oriented L2CAP channel, whereas point-to-multipoint signalling is mapped towards the connectionless L2CAP channel, which in turn is sent as broadcast information on the beacon channel (piconet broadcast).

Figure 1.2 illustrates point-to-point signalling to establish a voice or data call in a single-point configuration. First the other device is notified of the call request using the point-to-point signalling channel (A). Next, this signalling channel is used to further establish the speech or data channel (B).

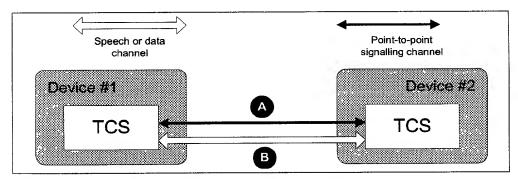


Figure 1.2: Point-to-point signalling in a single-point configuration

Figure 1.3 below illustrates how point-to-multipoint signalling and point-to-point signalling is used to establish a voice or data call in a multi-point configuration. First all devices are notified of the call request using point-to-multipoint signalling channel (A). Next, one of the devices answers the call on the point-to-point signalling channel (B); this signalling channel is used to further establish the speech or data channel (C).

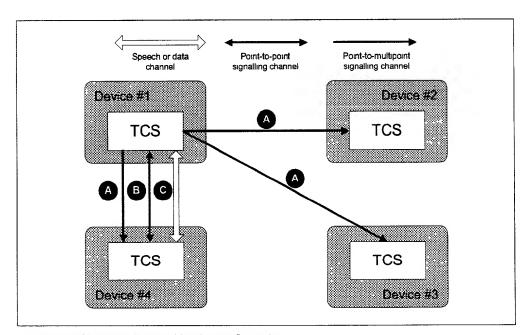


Figure 1.3: Signalling in a multi-point configuration

Bluetooth.

1.3 OPERATION BETWEEN LAYERS

TCS implementations should follow the general architecture described below (note that, for simplicity, handling of data calls is not drawn).

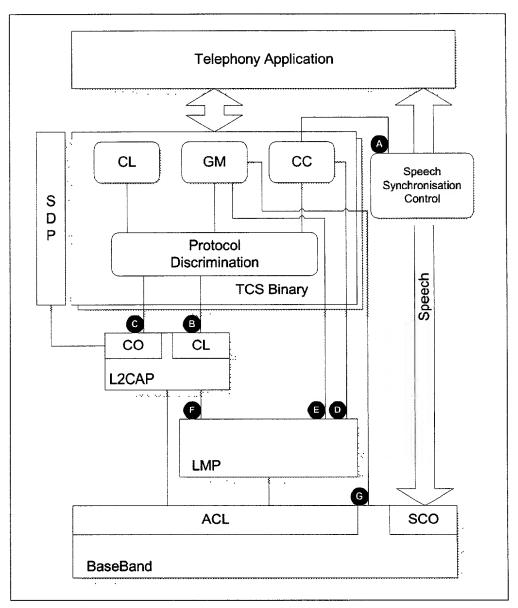


Figure 1.4: TCS Architecture

The internal structure of TCS Binary contains the functional entities Call Control, Group Management and ConnectionLess as described in Section 1.1 on page 435, complemented with the Protocol Discrimination which, based upon the TCS internal protocol discriminator, routes traffic to the appropriate functional entity.

Bluetooth.

To handle more calls simultaneously, multiple instances of TCS Binary may exist at the same time. Discrimination between the multiple instances can be based on the L2CAP channel identifier.

TCS Binary interfaces with a number of other (Bluetooth) entities to provide its (telephone) services to the application. The interfaces are identified in Figure 1.4 above, and information is exchanged across these interfaces for the following purposes:

- A The Call Control entity provides information to the speech synchronization control about when to connect (disconnect) the speech paths. This information is based upon the call control messages (e.g. reception of CONNECT ACKNOWLEDGE or DISCONNECT, see Section 2 on page 439)
- B To send a SETUP message (see Section 2.2.1 on page 439) using point-to-multipoint signalling, it is delivered on this interface to L2CAP for transmission on the connectionless channel. The other way round L2CAP uses this interface to inform TCS of a SETUP message received on the connectionless channel. The connectionless L2CAP channel maps onto the piconet broadcast
- C Whenever a TCS message needs to be sent using point-to-point signalling, it is delivered on this interface to L2CAP for transmission on a connection-oriented channel. During L2CAP channel establishment specific quality of service to be used for the connection will be indicated, in particular the usage of low power modes (L2CAP will inform LMP about this interface F)
- D The Call Control entity controls the LMP directly, for the purpose of establishing and releasing SCO links
- E & G. The Group Management entity controls the LMP and LC/Baseband directly during initialization procedures to control (for example) the inquiry, paging and pairing.

Bluetooth.

2 CALL CONTROL (CC)

2.1 CALL STATES

The call states used by the TCS are those identified in Q.931[1], for the user side only. To allow for implementation within computing power- and memory-restricted devices, only a subset of the states is mandatory for TCS based implementations. This mandatory subset is termed **Lean TCS**.

The states are named as follows. States in bold are mandatory states, part of Lean TCS:

General States

Null (0)

Active (10)

Disconnect request (11)

Disconnect indication (12)

Release request (19)

Outgoing Side States

Call initiated (1)

Overlap sending (2)

Outgoing call proceeding (3)

Call delivered (4)

Incoming Side States

Call present (6)

Call received (7)

Connect request (8)

Incoming call proceeding (9)

Overlap receiving (25)

These states, together with the state transitions, have been indicated in the state diagram contained in Appendix 1 – TCS Call States. For clarity, a separate state diagram has been included for Lean TCS.

2.2 CALL ESTABLISHMENT

A connection-oriented L2CAP channel between the Outgoing and Incoming Side shall be available before any of the CC procedures can operate.

Additionally, in a multi-point configuration (see Section 1.2 on page 435), a connectionless L2CAP channel shall be available between the Outgoing and Incoming Side.

2.2.1 Call Request

The Outgoing Side initiates call establishment by sending a SETUP message, and starting timer T303.

Bluetooth.

In case of a single-point configuration (see Section 1.2 on page 435), the SETUP message is delivered on the connection-oriented channel.

In case of a multi-point configuration (see Section 1.2 on page 435), the SETUP message may be delivered on the connection-less channel. This causes the SETUP message to be transmitted as a broadcast message at every beacon instant (as described in Baseband Specification Section 10.8.4 on page 115).

If no response (as prescribed in Section 2.2.4 on page 441) is received from the Incoming Side before timer T303 expires, the Outgoing Side shall:

- 1. If the SETUP message was delivered on a connection-less channel, return to the Null state. This stops the transmission of the SETUP message.
- 2. If the SETUP message was delivered on a connection-oriented channel, send a RELEASE COMPLETE message to the Incoming Side. This message should contain cause # 102, recovery on timer expiry.

The SETUP message shall always contain the call class. It shall also contain all the information required by the Incoming Side to process the call. The number digits within the Called party number information element may optionally be incomplete, thus requiring the use of overlap sending (Section 2.2.3 on page 441). The SETUP message may optionally contain the Sending complete information element in order to indicate that the number is complete.

Following the transmission of the SETUP message, the Outgoing Side shall enter the Call initiated state. On receipt of the SETUP message the Incoming Side shall enter the Call present state.

2.2.2 Bearer selection

440

The SETUP message sent during the Call Request may contain the Bearer capability information element, to indicate the requested bearer. The Incoming Side may negotiate on the requested bearer by including a Bearer capability information element in the first message in response to the SETUP message.

The Bearer capability information element indicates which lower layer resources (the *bearer channel*) are used during a call. If bearer capability 'Synchronous Connection-Oriented (SCO)' is indicated, an SCO link will be used, with the indicated packet type and voice coding to enable speech calls. If bearer capability 'Asynchronous Connection-Less (ACL)' is indicated, an ACL link will be used. On top of this, there will be an L2CAP channel with indicated QoS requirements, to enable data calls. If bearer capability 'None' is indicated, no separate bearer channel will be established.

Note: it is the responsibility of the implementation to assure that the bearer capability as indicated is available to the call.

Bluetooth.

2.2.3 Overlap Sending

If the received SETUP message does not contain a Sending complete indication information element, and contains either –

- a) incomplete called-number information, or
- b) called-number information which the Incoming Side cannot determine to be complete,

then the Incoming Side shall start timer T302, send a SETUP ACKNOWL-EDGE message to the Outgoing Side, and enter the Overlap receiving state.

When the SETUP ACKNOWLEDGE message is received, the Outgoing Side shall enter the Overlap sending state, stop timer T303, and start timer T304.

After receiving the SETUP ACKNOWLEDGE message, the Outgoing Side shall send the remainder of the call information (if any) in the called party number information element of one or more INFORMATION messages.

The Outgoing Side shall restart timer T304 when each INFORMATION message is sent.

The INFORMATION message, which completes the information sending, may contain a sending complete information element. The Incoming Side shall restart timer T302 on receipt of every INFORMATION message not containing a sending complete indication, if it cannot determine that the called party number is complete.

At the expiry of timer T304, the Outgoing Side shall initiate call clearing in accordance with Section 2.3.1 with cause #102, recovery on timer expiry.

At the expiry of timer T302, the Incoming Side shall:

- if it determines that the call information is incomplete, initiate call clearing in accordance with Section 2.3.1 with cause #28, invalid number format.
- otherwise the Incoming Side shall reply with a CALL PROCEEDING, ALERTING or CONNECT message.

2.2.4 Call Proceeding

2.2.4.1 Call proceeding, enbloc sending

If enbloc sending is used (i.e. the Incoming Side can determine it has received sufficient information in the SETUP message from the Outgoing Side to establish the call) the Incoming Side shall send a CALL PROCEEDING message to the Outgoing Side to acknowledge the SETUP message and to indicate that the call is being processed. Upon receipt of the CALL PROCEEDING message, the Outgoing Side shall enter the Outgoing Call proceeding state stop

Bluetooth.

timer T303 and start timer T310. After sending the CALL PROCEEDING message, the Incoming Side shall enter the Incoming Call proceeding state.

2.2.4.2 Call proceeding, overlap sending

Following the occurrence of one of these conditions -

- the receipt by the Incoming Side of a Sending complete indication, or
- analysis by the Incoming Side that all call information necessary to effect call establishment has been received,

the Incoming Side shall send a CALL PROCEEDING message to the Outgoing Side, stop timer T302, and enter the Incoming Call proceeding state.

When the Outgoing Side receives of the CALL PROCEEDING message it shall enter the Outgoing Call proceeding state, stop timer T304 and, if applicable, start timer T310.

2.2.4.3 Expiry of timer T310

On expiry of T310 (i.e. if the Outgoing Side does not receive an ALERTING, CONNECT, DISCONNECT or PROGRESS message), the Outgoing Side shall initiate call clearing in accordance with Section 2.3.1 on page 446 with cause #102, recovery on timer expiry.

2.2.5 Call Confirmation

Upon receiving an indication that user alerting has been initiated at the called address, the Incoming Side shall send an ALERTING message, and shall enter the Call received state.

When the Outgoing Side receives the ALERTING message, the Outgoing Side may begin an internally generated alerting indication and shall enter the Call delivered state. The Outgoing Side shall stop timer T304 (in case of overlap receiving), stop timer T303 or T310 (if running), and start timer T301 (unless another internal altering supervision timer function exists).

On expiry of T301, the Outgoing Side shall initiate call clearing in accordance with Section 2.3.1 on page 446 with cause #102, recovery on timer expiry.

2.2.6 Call Connection

An Incoming Side indicates acceptance of an incoming call by sending a CONNECT message to the Outgoing Side, and stopping the user alerting. Upon sending the CONNECT message the Incoming Side shall start timer T313.

Bluetooth.

On receipt of the CONNECT message, the Outgoing Side shall stop any internally generated alerting indications, shall stop (if running) timers T301, T303, T304, and T310, shall complete the requested bearer channel to the Incoming Side, shall send a CONNECT ACKNOWLEDGE message, and shall enter the Active state.

The CONNECT ACKNOWLEDGE message indicates completion of the requested bearer channel. Upon receipt of the CONNECT ACKNOWLEDGE message, the Incoming Side shall connect to the bearer channel, stop timer T313 and enter the Active state.

When timer T313 expires prior to the receipt of a CONNECT ACKNOWLEDGE message, the Incoming Side shall initiate call clearing in accordance with Section 2.3.1 on page 446 with cause #102, recovery on timer expiry.

2.2.7 Call Information

While in the Active state, both sides may exchange any information related to the ongoing call using INFORMATION messages.

2.2.8 Non-selected user clearing

When the call has been delivered on a connection-less channel (in case of a multi-point configuration), in addition to sending a CONNECT ACKNOWL-EDGE message to the Incoming Side selected for the call, the Outgoing Side shall send a RELEASE message (indicating cause #26, non-selected user clearing) to all other Incoming Sides that have sent SETUP ACKNOWLEDGE, CALL PROCEEDING, ALERTING, or CONNECT messages in response to the SETUP message. These RELEASE messages are used to notify the Incoming Sides that the call is no longer offered to them.

2.2.9 In-band tones and announcements

When the Incoming Side provides in-band tones/announcements, and if the requested bearer implies speech call, the Incoming Side will first complete the bearer channel (if not already available). Then a progress indicator #8, *in-band information or appropriate pattern is now available* is sent simultaneously with the application of the in-band tone/announcement. This progress indicator may be included in any call control message that is allowed to contain the progress indicator information element or, if there is no call state change, in a dedicated PROGRESS message.

Upon receipt of this message, the Outgoing Side may connect (if not already connected) to the bearer channel to receive the in-band tone/announcement.

Bluetooth.

2.2.10 Failure of call establishment

In the Call present, Overlap receiving, Incoming call proceeding, or Call received states, the Incoming Side may initiate clearing as described in Section 2.3 on page 446 with a cause value indicated. Examples of some the cause values that may be used to clear the call, when the Incoming Side is in the Call present, Overlap receiving, or Incoming call proceeding state are the following:

```
#1 unassigned (unallocated) number
#3 no route to destination
#17 user busy
#18 no user responding
#22 number changed
#28 invalid number format (incomplete number)
#34 no circuit/channel available
#44 requested circuit/channel not available
#58 bearer capability not presently available
#65 bearer capability not implemented
```

Examples of two of the cause values that may be used to clear the call when the Incoming Side is in the Call received state are as follows:

```
#19 no answer from user (user alerted)
#21 call rejected by user
```

Bluetooth.

2.2.11 Call Establishment Message Flow

The figure below provides a complete view of the messages exchanged during successful Call Establishment, as described in the sections above. The mandatory messages, part of the Lean TCS, are indicated by a solid arrow. A dotted arrow indicates the optional messages. A triangle indicates a running timer.

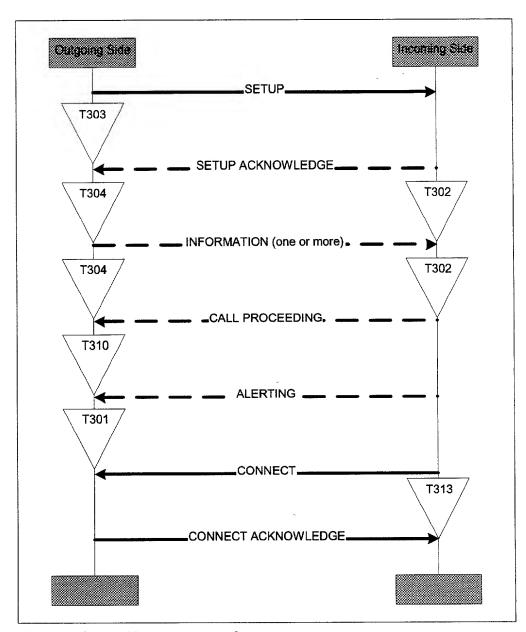


Figure 2.1: Call establishment message flow

Bluetooth.

2.3 CALL CLEARING

2.3.1 Normal Call Clearing

Apart from the exceptions identified in Section 2.3.2 on page 447, the clearing procedures are symmetrical and may be initiated by either the Outgoing or the Incoming Side. In the interest of clarity, the following procedures describe only the case where the Outgoing Side initiates clearing.

On sending or receiving any call clearing message, any protocol timer other than T305 and T308 shall be stopped.

The Outgoing Side shall initiate clearing by sending a DISCONNECT message, starting timer T305, disconnecting from the bearer channel, and entering the Disconnect request state.

The Incoming Side shall enter the Disconnect indication state upon receipt of a DISCONNECT message. This message prompts the Incoming Side to disconnect from the bearer channel. Once the channel used for the call has been disconnected, the Incoming Side shall send a RELEASE message to the Outgoing Side, start timer T308, and enter the Release request state.

On receipt of the RELEASE message the Outgoing Side shall cancel timer T305, release the bearer channel, send a RELEASE COMPLETE message, and return to the Null state.

Following the receipt of a RELEASE COMPLETE message from the Outgoing Side, the Incoming Side shall stop timer T308, release the bearer channel, and return to the Null state.

If the Outgoing Side does not receive a RELEASE message in response to the DISCONNECT message before timer T305 expires, it shall send a RELEASE message to the Incoming Side with the cause number originally contained in the DISCONNECT message, start timer T308 and enter the Release request state.

If in the Release request state, a RELEASE COMPLETE message is not received before timer T308 expires, the side that expected the message shall return to the Null state.

Clearing by the called user employing user-provided tones/announcements

In addition to the procedures described above, if the requested bearer signals a speech call, the Outgoing Side may apply in-band tones/announcements in the clearing phase. When in-band tones/announcements are provided, the Outgoing Side will first complete the bearer channel (if not already available), and next send the DISCONNECT message containing progress indicator #8, in-band information or appropriate pattern is now available.

Bluetooth.

Upon receipt of this message, the Incoming Side may connect (if not already connected) to the bearer channel to receive the in-band tone/announcement, and enter the Disconnect indication state.

The Incoming Side may subsequently continue clearing (before the receipt of a RELEASE from the Outgoing Side) by disconnecting from the bearer channel, sending a RELEASE message, starting timer T308, and entering the Release request state.

2.3.2 Abnormal Call Clearing

Under normal conditions, call clearing is initiated when either side sends a DIS-CONNECT message and follows the procedures defined in Section 2.3.1 on page 446. The only exceptions to the above rule are as follows:

- a In response to a SETUP message, the Incoming Side can reject a call (e.g. because of unavailability of suitable resources) by responding with a RELEASE COMPLETE message provided no other response has previously been sent, and enter the Null state
- b In case of a multi-point configuration, non-selected user call clearing will be initiated with RELEASE message(s) from the Outgoing Side (Section 2.2.8 on page 443)
- c In case of a multi-point configuration, where the SETUP message is delivered on an connection-less channel, if a remote (calling) user disconnect indication is received during call establishment, any Incoming Side which has responded, or subsequently responds, shall be cleared by a RELEASE message, and the procedures of Section 2.3.1 on page 446 are then followed for that user. The Outgoing Side enters the Null state upon completion of clearing procedures for all responding Incoming Sides.

2.3.3 Clear Collision

Clear collision occurs when the Incoming and the Outgoing Sides simultaneously transfer DISCONNECT messages. When either side receives a DISCONNECT message while in the Disconnect request state, the side shall stop timer T305, disconnect the bearer channel (if not disconnected), send a RELEASE message, start timer T308, and enter the Release request state.

Clear collision can also occur when both sides simultaneously transfer RELEASE messages. The entity receiving such a RELEASE message while within the Release request state shall stop timer T308, release the bearer channel, and enter the Null state (without sending or receiving a RELEASE COMPLETE message).

Bluetooth.

2.3.4 Call Clearing Message Flow

The figure below provides the complete view on the messages exchanged during normal Call Clearing, as described in the sections above. All messages are mandatory.

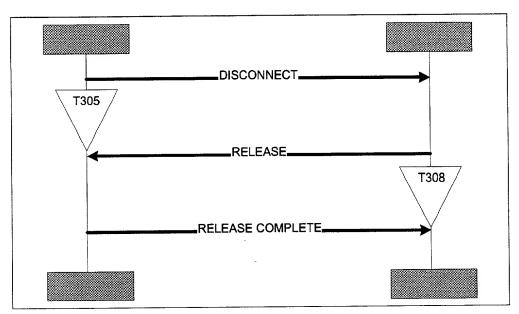


Figure 2.2: Call clearing message flow

Bluetooth.

3 GROUP MANAGEMENT (GM)

3.1 OVERVIEW

The Group Management entity provides procedures for managing a group of devices.

The following procedures are supported:

- Obtain access rights (Section 3.3 on page 451)
 enables the requesting device to use the telephony services
 of another device, part of a group of devices
- Configuration distribution (Section 3.4 on page 452)
 facilitates the handling and operation of a group of devices
- Fast inter-member access (Section 3.5 on page 453)
 enables faster contact establishment between devices of the same group

A connection-oriented L2CAP channel between devices shall be available before any of the GM procedures can operate.

For group management, the concept of Wireless User Group (WUG) is used.

3.2 THE WIRELESS USER GROUP

3.2.1 Description

A WUG consists of a number of Bluetooth units supporting TCS. One of the devices is called the WUG master. The WUG master is typically a gateway, providing the other Bluetooth devices – called WUG members – with access to an external network. All members of the WUG in range are members of a piconet (active or parked). Master of this piconet is always the WUG master.

The main relational characteristics of a WUG are:

- All units that are part of a WUG know which unit is the WUG master and which other units are member of this WUG. WUG members receive this information from the WUG master.
- When a new unit has paired with the WUG master, it is able to communicate
 and perform authentication and encryption with any other unit part of the
 WUG without any further pairing/initialization. The WUG master provides the
 required authentication and encryption parameters to the WUG members.

Both relational characteristics are maintained through the Configuration distribution procedure.

Bluetooth.

3.2.2 Encryption within the WUG

In order to allow for encrypted transmission on the connectionless L2CAP channel, the WUG master issues a temporary key (K_{master}). As a Bluetooth unit is not capable of switching between two or more encryption keys in real time, this key is normally also used for encrypted transmission on the connection-oriented channel (individually addressed traffic). Since the WUG master piconet may be in operation for extended periods without interruption, the K_{master} shall be changed periodically.

In order to allow for authentication and encryption to be performed between WUG members, the WUG master may use the Configuration distribution procedure to issue link keys that the WUG members use for communication with each other. Just as if pairing had created these keys, the keys are unique to a pair of WUG members and hence a WUG member uses a different key for every other WUG member it connects to.

The Configuration distribution shall always be performed using encrypted links. The K_{master} shall not be used for encryption; rather the WUG master shall ensure that the semi-permanent key for the specific WUG member addressed shall be used.

3.2.3 Unconscious pairing

For TCS, pairing a device with the WUG master implies pairing a device with all members of the WUG. This is achieved using the Configuration distribution procedure. This avoids the user of the device having to pair with each and every device of the WUG individually.

In Bluetooth, pairing is not related to a specific service but rather to a specific device. After pairing, all services provided by a device are accessible, if no further application- or device-specific protection is provided.

Without further provisions, pairing a device with the WUG master implies that all services provided by the new device are accessible to all other WUG members. And vice versa, without further provisions, the new device can access all services provided by other WUG members.

For this reason, implementers of TCS – and in particular the Configuration distribution procedure – are recommended to add provisions where:

- 1. a new device entering the WUG is not mandated to initiate the Obtain access rights procedure to become a WUG member, and is consequently only able to use the services provided by the WUG master (gateway)
- 2. a WUG master can reject a request to obtain access rights
- 3. a WUG member is not forced to accept the pairing information received during the Configuration distribution

This applies in particular to devices offering more than just TCS- related services.

Bluetooth.

3.3 OBTAIN ACCESS RIGHTS

Using the Obtain access rights procedure, a device can obtain the rights to use the telephony services provided by another device, part of a WUG.

3.3.1 Procedure description

A device requests access rights by sending an ACCESS RIGHTS REQUEST message and starting timer T401. Upon receipt of the ACCESS RIGHTS REQUEST message, the receiving device accepts the request for access rights by sending an ACCESS RIGHTS ACCEPT.

When the requesting device receives the ACCESS RIGHTS ACCEPT, it shall stop timer T401. Then, the access rights procedure has completed successfully.

If no response has been received before the expiration of timer T401, the requesting device shall consider the request for access rights to be denied.

If, upon receipt of the ACCESS RIGHTS REQUEST message, the receiving device is for some reason unable to accept the access rights, it shall reply with an ACCESS RIGHTS REJECT message. Upon receipt of an ACCESS RIGHTS REJECT message, the requesting device shall stop timer T401 and consider the request for access rights to be denied.

3.3.2 Message flow

The figure below provides the complete view on the messages exchanged during the Obtain access rights procedure.

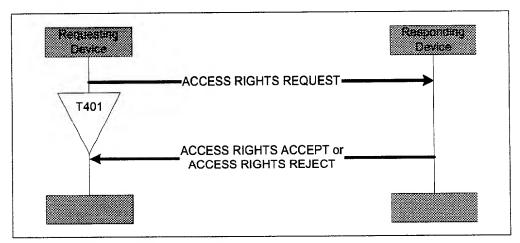


Figure 3.1: Obtain access rights message flow

Bluetooth.

3.4 CONFIGURATION DISTRIBUTION

The units in the WUG need to be informed about changes in the WUG; e.g. when a unit is added or removed. The Configuration distribution procedure is used to exchange this data.

When a WUG configuration change occurs, the WUG master initiates the Configuration distribution procedure on all WUG members. The WUG master keeps track of which WUG members have been informed of WUG configuration changes.

Some WUG members may be out of range and may therefore not be reached. The update of these WUG members will be performed when these members renew contact with the WUG master.

3.4.1 Procedure Description

The WUG master initiates the Configuration distribution procedure by starting timer T403, and transferring the INFO SUGGEST message. The INFO SUGGEST message contains the complete WUG configuration information. Upon receipt of the INFO SUGGEST message, the WUG member shall send an INFO ACCEPT message, to acknowledge the proper receipt of the WUG configuration information.

When the WUG master receives the INFO ACCEPT, the timer T401 is stopped, and the Configuration distribution procedure has completed successfully. On expiry of timer T403, the Configuration distribution procedure is terminated.

3.4.2 Message flow

The figure below provides the complete view on the messages exchanged during the Configuration distribution procedure, as described in the sections above.

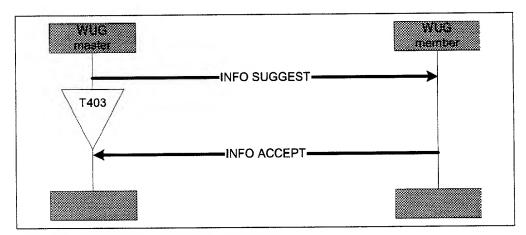


Figure 3.2: Configuration distribution message flow

Bluetooth.

3.5 FAST INTER-MEMBER ACCESS

When two WUG members are both active in the WUG master piconet, a WUG member can use the Fast inter-member access procedure to obtain fast access to another WUG member. With the Fast inter-member access procedure, the originating WUG member obtains clock information from the terminating WUG member and forces the terminating WUG member to go into PAGE_SCAN for a defined period (T406).

3.5.1 Listen Request

The originating WUG member initiates the Fast inter-member access procedure by starting timer T404 and transferring the LISTEN REQUEST message to the WUG master, indicating the WUG member with which it wishes to establish contact.

If, before expiry of timer T404, the originating WUG member receives no response to the LISTEN REQUEST message, the originating WUG member shall terminate the procedure.

3.5.2 Listen Accept

Upon receipt of the LISTEN REQUEST message, the WUG master checks that the indicated WUG member is part of the WUG. If this is the case, the WUG master initiates the Fast inter-member access towards the terminating WUG member side by starting timer T405 and sending the LISTEN SUGGEST message to the terminating WUG member.

Upon receipt of the LISTEN SUGGEST message, the terminating WUG member confirms the suggested action (internal call) by sending a LISTEN ACCEPT message to the WUG master. This message contains the terminating WUG member's clock offset. After sending the LISTEN ACCEPT, the terminating WUG member shall go to PAGE-SCAN state, for T406 seconds, to enable connection establishment by the originating WUG member.

Upon receipt of the LISTEN ACCEPT message, the WUG master stops timer T405, and informs the originating WUG member of the result of the WUG fast inter-member access by sending a LISTEN ACCEPT message. This message contains the terminating WUG member's clock offset. Upon receipt of the LISTEN ACCEPT message, the originating WUG member stops timer T404, and starts paging the terminating WUG member.

If no response to the LISTEN SUGGEST message is received by the WUG master before the first expiry of timer T405, then the WUG master shall terminate the Fast inter-member access procedure by sending a LISTEN REJECT message to both originating and terminating WUG member using cause #102, recovery on timer expiry.

Bluetooth.

3.5.3 Listen Reject by the WUG Master

If the WUG master rejects the Fast inter-member access procedure, it sends a LISTEN REJECT message to the originating WUG member.

Valid cause values are:

#1, *Unallocated (unassigned) number* (when the indicated WUG member is not part of the WUG)

#17, User busy (in case terminating WUG member is engaged in an external call)

#20, Subscriber absent (upon failure to establish contact with the terminating WUG member), or

any cause value indicated in a LISTEN REJECT message received from/sent to the terminating WUG member.

Upon receipt of the LISTEN REJECT message, the originating WUG member stops timer T404, and terminates the procedure.

3.5.4 Listen Reject by the WUG Member

If the terminating WUG member rejects the suggested action received in the LISTEN SUGGEST message, it sends a LISTEN REJECT message to the WUG master. Valid cause value is #17, *User busy* (in case terminating WUG member is engaged in another internal call).

Upon receipt of the LISTEN REJECT, the WUG master stops timer T405, and continues as described in Section 3.5.3 on page 454.

3.5.5 Message flow

The figure below provides a view of the messages exchanged during Fast intermember access, as described in the sections above. A successful Fast intermember access procedure ends with the terminating WUG member going into page scan, thus allowing the originating WUG member to contact him directly.

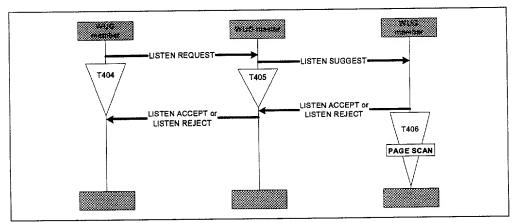


Figure 3.3: Fast inter-member access message flow

Bluetooth.

4 CONNECTIONLESS TCS (CL)

A connectionless TCS message can be used to exchange signalling information without establishing a TCS call. It is thus a connectionless service offered by TCS.

A connectionless TCS message is a CL INFO message (as defined in Section 6.3.1 on page 470).

A connection-oriented L2CAP channel between the Outgoing and Incoming Side shall be available before a CL INFO message can be sent.

Note: In the case of a connection-oriented channel, it may choose to delay the termination of the channel for a defined period to exchange more CL INFO messages.

Alternatively, in a multi-point configuration (see Section 1.2 on page 435), a connectionless L2CAP channel may be used and, if so, shall be available before a CL INFO can be sent.

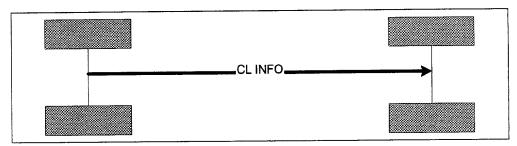


Figure 4.1: Connectionless TCS message flow

Bluetooth.

5 SUPPLEMENTARY SERVICES (SS)

The TCS provides explicit support for only one supplementary service, the Calling Line Identity (seeSection 5.1 on page 456).

For supplementary services provided by an external network, using DTMF sequences for the activation/de-activation and interrogation of supplementary services, the DTMF start & stop procedure is supported (see Section 5.2 on page 456). This procedure allows both finite and infinite tone lengths.

Section 5.3 on page 458 specifies how a specific supplementary service, provided by an external network, called register recall is supported.

For other means of supplementary service control, no explicit support is specified. Support may be realized by either using the service call, or use the company specific information element, or a combination.

5.1 CALLING LINE IDENTITY

To inform the Incoming Side of the identity of the originator of the call, the Outgoing Side may include the calling party number information element (see Section 7.4.6 on page 481) in the SETUP message transferred as part of the call request.

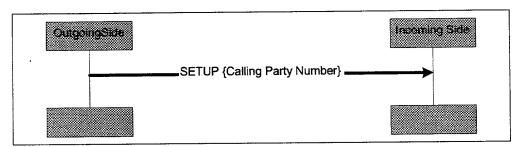


Figure 5.1: Calling line identity message flow

5.2 DTMF START & STOP

The DTMF start & stop procedure is supported to provide supplementary service control on PSTN type of networks.

In principle DTMF messages can be initiated by either (Outgoing or Incoming) Side; in practice, however, the Side (gateway) connected to the external PSTN network will be the recipient.

DTMF messages can be transmitted only in the active state of a call. Tone generation shall end when the call is disconnected.

Bluetooth.

5.2.1 Start DTMF request

A user may cause a DTMF tone to be generated; e.g. by depression of a key. The relevant action is interpreted as a requirement for a DTMF digit to be sent in a START DTMF message on an established signalling channel. This message contains the value of the digit to be transmitted (0, 1...9, A, B, C, D, *, #).

Only a single digit will be transferred in each START DTMF message.

5.2.2 Start DTMF response

The side receiving the START DTMF message will reconvert the received digit back into a DTMF tone which is applied toward the remote user, and return a START DTMF ACKNOWLEDGE message to the initiating side. This acknowledgment may be used to generate an indication as a feedback for a successful transmission.

If the receiving side cannot accept the START DTMF message, a START DTMF REJECT message will be sent to the initiating side, using cause #29, Facility rejected, indicating that sending DTMF is not supported by the external network.

5.2.3 Stop DTMF request

When the user indicates the DTMF sending should cease (e.g. by releasing the key) the initiating side will send a STOP DTMF message to the other side.

5.2.4 Stop DTMF response

Upon receiving the STOP DTMF message, the receiving side will stop sending the DTMF tone (if still being sent) and return a STOP DTMF ACKNOWLEDGE message to the initiating side.

5.2.5 Message flow

The figure below provides a view of the messages exchanged when a single DTMF tone needs to be generated.

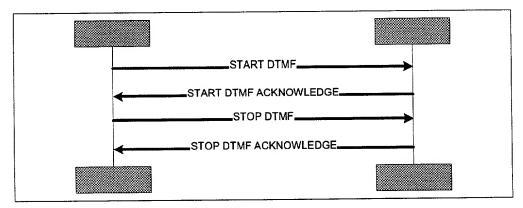


Figure 5.2: DTMF start & stop message flow

Bluetooth.

5.3 REGISTER RECALL

Register recall means to seize a register (with dial tone) to permit input of further digits or other action. In some markets, this is referred to as 'hook flash'. Register recall is supported by sending an INFORMATION message with a keypad facility information element, indicating 'register recall' (value 16H). Further digits are sent using the procedures as indicated in Section 5.2 above.

Bluetooth.

6 MESSAGE FORMATS

This section provides an overview of the structure of messages used in this specification, and defines the function and information contents (i.e. semantics) of each message.

Whenever a message is sent according to the procedures of Sections 2, 3 and 4, it shall contain the mandatory information elements, and optionally any combination of the optional information elements specified in this section for that message.

A message shall always be delivered in a single L2CAP packet. The start of a message (the LSB of the first octet) shall be aligned with the start of the L2CAP payload.

Each definition includes:

- a) A brief description of the message direction and use
- b) A table listing the information elements in order of their appearance in the message (same relative order for all message types)
- c) Indications for each information element in the table, specifying -
 - · the section of this specification describing the information element
 - · whether inclusion in mandatory ('M') or optional ('O')
 - the length (or length range) of the information element, where '*'
 denotes an undefined maximum length which may be application
 dependent.
- d) Further explanatory notes, as necessary

All message formats are denoted in octets.

Bluetooth.

6.1 CALL CONTROL MESSAGE FORMATS

6.1.1 ALERTING

This message is sent by the incoming side to indicate that the called user alerting has been initiated.

Message Type: ALERTING

Direction: incoming to outgoing

Information Element	Ref.	Type	Length
Message type	7.3	М	1
Bearer capability	7.4.3	O Note 1)	4(26)
Progress indicator	7.4.13	0	2
SCO Handle	7.4.14	0	2
Destination CID	7.4.11	0	4
Company specific	7.4.9	0	3-*

Table 6.1: ALERTING message content

Note 1: Allowed only in the first message sent by the incoming side.

6.1.2 CALL PROCEEDING

This message is sent by the incoming side to indicate that the requested call establishment has been initiated and no more call establishment information will be accepted.

Message Type: CALL PROCEEDING

Direction: incoming to outgoing

460

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Bearer capability	7.4.3	O Note 1)	4(26)
Progress indicator	7.4.13	0	2
SCO Handle	7.4.14	0	2
Destination CID	7.4.11	0	4
Company specific	7.4.9	0	3-*

Table 6.2: CALL PROCEEDING message content

Note 1: Allowed only in the first message sent by the incoming side.

Bluetooth.

6.1.3 CONNECT

This message is sent by the incoming side to indicate call acceptance by the called user.

Message Type: CONNECT

Direction: incoming to outgoing

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Bearer capability	7.4.3	O ^{Note 1)}	4(26)
SCO Handle	7.4.14	0	2
Company specific	7.4.9	0	3-*

Table 6.3: CONNECT message content

Note 1: Allowed only in the first message sent by the incoming side.

6.1.4 CONNECT ACKNOWLEDGE

This message is sent by the outgoing side to acknowledge the receipt of a CONNECT message.

Message Type: CONNECT ACKNOWLEDGE

Direction: outgoing to incoming

Information Element	Ref.	Type	Length
Message type	7.3	М	1
SCO Handle	7.4.14	O	2
Destination CID	7.4.11	0	4
Company specific	7.4.9	o	3-*

Table 6.4: CONNECT ACKNOWLEDGE message content

Bluetooth.

6.1.5 DISCONNECT

This message is sent by either side as an invitation to terminate the call.

Message Type: DISCONNECT

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	M	1
Cause	7.4.7	0	2
Progress indicator	7.4.13	0	2
SCO Handle	7.4.14	0	2
Destination CID	7.4.11	0	4
Company specific	7.4.9	0	3-*

Table 6.5: DISCONNECT message content

6.1.6 INFORMATION

This message is sent by either side to provide additional information during call establishment (in case of overlap sending).

Message Type: INFORMATION

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	M	1
Sending complete	7.4,15	0	1
Keypad facility	7.4.12	0	2
Called party number	7.4.5	0	3-*
Audio control	7.4.2	О	3-*
Company specific	7.4,9	O	3-*

Table 6.6: INFORMATION message content

Bluetooth.

6.1.7 PROGRESS

This message is sent by the incoming side to indicate the progress of a call in the event of interworking or by either side in the call with the provision of optional in-band information/patterns.

Message Type: PROGRESS

Direction: incoming to outgoing

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Progress indicator	7.4.13	М	2
SCO Handle	7.4.14	0	2
Destination CID	7.4.11	o	4
Company specific	7.4.9	0	3-*

Table 6.7: PROGRESS message content

6.1.8 RELEASE

This message is used to indicate that the device sending the message had disconnected the channel (if any) and intends to release the channel, and that receiving device should release the channel after sending RELEASE COMPLETE.

Message Type: RELEASE

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	M	1
Cause	7.4,7	O ^{Nate 1)}	2
Company specific	7.4.9	0	3-*

Table 6.8: RELEASE message content

Note 1: Mandatory in the first call clearing message.

Bluetooth.

6.1.9 RELEASE COMPLETE

This message is used to indicate that the device sending the message has released the channel (if any), and that the channel is available for re-use.

Message Type: RELEASE COMPLETE

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	М	1
Cause	7.4.7	Q ^{Note 1)}	2
Company specific	7.4.9	0	3-*

Table 6.9: RELEASE COMPLETE message content

Note 1: Mandatory in the first call clearing message.

6.1.10 SETUP

This message is sent by the outgoing side to initiate call establishment.

Message Type:

Direction:

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Call class	7.4.4	M	2
Sending complete	7.4.15	0	1
Bearer capability	7.4.3	0	4(26)
Signal	7.4.16	0	2
Calling party number	7.4.6	0	3-*
Called party number	7.4.5	0	3-*
Company specific	7.4.9	0	3-*

Table 6.10: SETUP message content

Bluetooth.

465

6.1.11 SETUP ACKNOWLEDGE

This message is sent by the incoming side to indicate that call establishment has been initiated, but additional information may be required.

Message Type: SETUP ACKNOWLEDGE

Direction: incoming to outgoing

Information Element	Ref.	Туре	Length
Message type	7.3	M	1
Bearer capability	7.4.3	O ^{Note 1)}	4(26)
Progress indicator	7.4.13	0	2
SCO Handle	7.4.14	0	2
Destination CID	7.4.11	0	4
Company specific	7.4.9	0	3-*

Table 6.11: SETUP ACKNOWLEDGE message content

Note 1: Allowed only in the first message sent by the incoming side.

6.1.12 Start DTMF

This message contains the digit the other side should reconvert back into a DTMF tone, which is then applied towards the remote user.

Message Type: Start DTMF

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Keypad facility	7.4.12	M	2

Table 6.12: Start DTMF message content

Bluetooth.

6.1.13 Start DTMF Acknowledge

This message is sent to indicate the successful initiation of the action required by the Start DTMF message.

Message Type: Start DTMF Acknowledge

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Keypad facility	7.4.12	M	2

Table 6.13: Start DTMF Acknowledge message content

6.1.14 Start DTMF Reject

This message is sent to indicate that the other side cannot accept the Start DTMF message.

Message Type: Start DTMF Reject

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Cause	7.4.7	0	2

Table 6.14: Start DTMF Reject message content

6.1.15 Stop DTMF

This message is used to stop the DTMF tone sent towards the remote user.

Message Type: Stop DTMF

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	М	1 `

Table 6.15: Stop DTMF message content

Message formats

Bluetooth.

6.1.16 Stop DTMF Acknowledge

This message is sent to indicate that the sending of the DTMF tone has been stopped.

Message Type: Stop DTMF Acknowledge

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Keypad facility	7.4.12	M	2

Table 6.16: Stop DTMF Acknowledge message content

6.2 GROUP MANAGEMENT MESSAGE FORMATS

6.2.1 ACCESS RIGHTS REQUEST

This message is sent by the initiating side to obtain access rights.

Message Type: ACCESS RIGHTS REQUEST

Direction:

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Company specific	7.4.9	0	3-*

Table 6.17: ACCESS RIGHTS REQUEST message content

6.2.2 ACCESS RIGHTS ACCEPT

This message is sent by the responding side to indicate granting of access rights.

Message Type: ACCESS RIGHTS ACCEPT

Direction:

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Company specific	7.4.9	0	3-*

Table 6.18: ACCESS RIGHTS ACCEPT message content

Bluetooth.

6.2.3 ACCESS RIGHTS REJECT

This message is sent by the responding side to indicate denial of access rights.

Message Type: ACCESS RIGHTS REJECT

Direction:

Information Element	Ref.	Туре	Length
Message type	7.3	M	1
Company specific	7.4.9	٥	3_*

Table 6.19: ACCESS RIGHTS REJECT message content

6.2.4 INFO SUGGEST

This message is sent by the WUG master to indicate that a change has occurred in the WUG configuration.

Message Type: INFO SUGGEST

Direction: WUG master to WUG member

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Configuration Data	7.4.10	M	*
Company specific	7.4.9	0	3-*

Table 6.20: INFO SUGGEST message content

6.2.5 INFO ACCEPT

This message is sent by the WUG member to indicate the acceptance of the updated WUG configuration.

Message Type: INFO ACCEPT

Direction: WUG member to WUG master

Information Element	Ref.	Type	Length
Message type	7.3	М	1
Company specific	7.4.9	0	3-*

Table 6.21: INFO ACCEPT message content

Bluetooth.

6.2.6 LISTEN REQUEST

This message is sent by a WUG member to indicate to the WUG master the request for a Fast inter-member access to the indicated WUG member.

Message Type: LISTEN REQUEST

Direction: WUG member to WUG master

Information Element	Ref.	Туре	Length
Message type	7.3	M	1
Called party number	7.4.6	M	3-*
Company specific	7.4.9	0	3-*

Table 6.22: LISTEN REQUEST message content

6.2.7 LISTEN SUGGEST

This message is sent by a WUG master to indicate to the WUG member the request for a Fast inter-member access.

Message Type: LISTEN SUGGEST

Direction: WUG master to WUG member

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Company specific	7.4.9	0	3-*

Table 6.23: LISTEN SUGGEST message content

6.2.8 LISTEN ACCEPT

This message is sent to indicate the acceptance of the previous request for a Fast inter-member access.

Message Type: LISTEN ACCEPT

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	М	1
Clock offset	7.4.8	0	4
Company specific	7.4.9	О	3-*

Table 6.24: LISTEN ACCEPT message content

Bluetooth.

6.2.9 LISTEN REJECT

This message is sent to indicate the rejection of the previous request for a Fast inter-member access.

Message Type: LISTEN REJECT

Direction: both

Information Element	Ref.	Type	Length
Message type	7.3	М	1
Cause	7.4.7	O	2
Company specific	7.4.9	0	3-*

Table 6.25: LISTEN REJECT message content

6.3 TCS CONNECTIONLESS MESSAGE FORMATS

6.3.1 CL INFO

This message is sent by either side to provide additional information in a connectionless manner.

Message Type: CL INFO

Direction: both

Information Element	Ref.	Туре	Length
Message type	7.3	M	1
Audio control	7.4.2	O	3-*
Company specific	7.4.9	0	3-*

Table 6.26: CL INFO message content

Bluetooth.

7 MESSAGE CODING

The figures and text in this section describe message contents. Within each octet, the bit designated 'bit 1' is transmitted first, followed by bit 2, 3, 4, etc. Similarly, the octet shown at the top of the figure is sent first.

Whenever a message is sent, according to the procedures of Sections 2, 3 and 4, it shall be coded as specified in this section.

7.1 OVERVIEW

The coding rules follow ITU-T Recommendation Q.931, but is tailored to the specific needs of TCS.

Every message consists of:

- a) Protocol discriminator
- b) Message type, and
- c) Other information elements, as required

The Protocol discriminator and Message type is part of every TCS message, while the other information elements are specific to each message type.

8		6	5	4	3	2	1	
Protocol	discrimina	itor	Message	type				octet 1
Other info	ormation e	elements a	s required					octet 2

Table 7.1: General message format

A particular information element shall be present only once in a given message.

The term 'default' implies that the value defined shall be used in the absence of any assignment or negotiation of alternative values.

For notation purposes – when a field extends over more than one octet, the order of bit values progressively decreases as the octet number increases. The least significant bit of the field is represented by the lowest numbered bit of the highest-numbered octet of the field. In general, bit 1 of each octet contains the least significant bit of a field.

Bluetooth.

7.2 PROTOCOL DISCRIMINATOR

The purpose of the protocol discriminator is to distinguish the TCS messages into different functional groups. The protocol discriminator is the first part of every message.

The protocol discriminator is coded according to Figure 7.1 and Table 7.2.

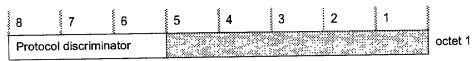


Figure 7.1: Protocol discriminator

Bi	ts		
8	7	6	
0	0	0	Bluetooth TCS Call Control
0	0	1	Bluetooth TCS Group management
0	1	Ō	Bluetooth TCS Connectionless
All	oth	er val	ues reserved

Table 7.2: Protocol discriminator

7.3 MESSAGE TYPE

The purpose of the message type is to identify the function of the message being sent.

The Message type is the first part of every message and it is coded as shown in Figure 7.2 and Table 7.3.

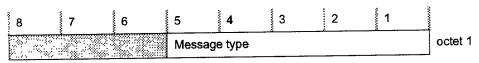


Figure 7.2: Message type

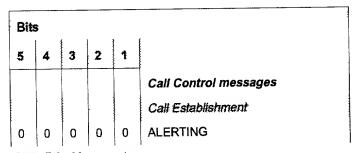


Table 7.3: Message type

Bluetooth.

Bit	S				_
5	4	3	2	1	
0	0	0	0	1	CALL PROCEEDING
0	0	0	1	0	CONNECT
Ø	0	0	1	1	CONNECT ACKNOWLEDGE
0	0	1	0	0	PROGRESS
O	0	1	0	1	SETUP
0	0	1	1	0	SETUP ACKNOWLEDGE
					Call clearing
0	0	1	1	1	DISCONNECT
0	1	0	O	0	RELEASE
0	1	0	0	1	RELEASE COMPLETE
			*******		Miscellaneous
0	1	0	1	0	INFORMATION
1	0	0	ø	O	START DTMF
1	0	0	0	1	START DTMF ACKNOWLEDGE
1	0	0	1	ø	START DTMF REJECT
1	0	0	1	1	STOP DTMF
1	0	1	0	Q	STOP DTMF ACKNOWLEDGE
					Group management messages
0	0	0	0	0	INFO SUGGEST
0	0	0	0	1	INFO ACCEPT
0	0	0	1	0	LISTEN REQUEST
0	0	0	1	1	LISTEN ACCEPT
O	0	1	O	0	LISTEN SUGGEST
0	0	1	0	1	LISTEN REJECT
O	0	1	1	0	ACCESS RIGHTS REQUEST
0	0	1	1	1	ACCESS RIGHTS ACCEPT
0	1	0	0	a	ACCESS RIGHTS REJECT
					Connectionless messages
0	0	0	0	0	CL INFO

Table 7.3: Message type

Bluetooth.

7.4 OTHER INFORMATION ELEMENTS

7.4.1 Coding rules

The coding of other information elements follows the coding rules described below.

Three categories of information elements are defined:

- a) single octet information elements (see Figure 7.3 on page 474)
- b) double octet information element (see Figure 7.4 on page 474)
- c) variable length information elements (see Figure 7.5 on page 474).

Table 7.4 on page 474 summarizes the coding of the information element identified bits for those information elements used in this specification.

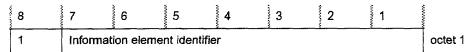


Figure 7.3: Single octet information element format

 8	7	6	5	4	3	2	1	***************************************
1	Infon	mation ele	ment iden	tifier				octet 1
Content	ts of inf	omation e	lement					octet 2

Figure 7.4: Double octet information element format

8	7	6	5	4	3	2	1					
0	0 Information element identifier											
Lengt	Length of contents of information element (octets)											
Conte	Contents of Information element											

Figure 7.5: Variable length information element format

	Coding								D-6	Max
8	7	6	5	4	3	2	1		Ref.	Length (octets)
1								Single octet information elements		
	0	1	0	0	0	0	1	Sending complete	7.4.15	1
1								Double octet information elements		

Table 7.4: Information element identifier coding

Bluetooth.

			Cod	ing					Ref.	Max Length
8	7	6	5	4	3	2	1			(octets)
	1	0	0	0	0	0	0	Call class	7.4.4	2
;	1	0	0	0	0	0	1	Cause	7.4.7	2
	1	0	0	0	O	1	0	Progress indicator	7.4.13	2
٠	1	0	0	0	0	1	1	Signal	7.4.16	2
	1	Q	Ð	0	1	0	Q	Keypad facility	7.4.12	2
	1	0	0	0	1	0	1	SCO handle	7.4.14	2
0								Variable length information elements	-	
	0	0	0	0	0	0	0	Clock offset	7.4.8	4
	0	0	0	0	Q	0	1	Configuration data	7.4.2	*
	0	0	0	0	0	1	0	Bearer capability	7.4.3	4(26)
	0	0	0	0	O	1	1	Destination CID	7,4,11	4
	0	0	0	0	1	0	0	Calling party number	7.4.6	*
	0	0	0	0	1	0	1	Called party number	7,4.5	*
	0	0	0	0	1	1	0	Audio control	7.4.2	*
	0	0	O	0	1	1	1	Company specific	7.4.9	*

Table 7.4: Information element identifier coding

The descriptions of the information elements below are organized in alphabetical order. However, there is a particular order of appearance for each information element in a message. The code values of the information element identifier for the variable length formats are assigned in ascending numerical order, according to the actual order of appearance of each information element in a message. This allows the receiving devices to detect the presence or absence of a particular information element without scanning through an entire message.

Where the description of information elements in this specification contains spare bits, these bits are indicated as being set to '0'. In order to allow compatibility with future implementation, messages should not be rejected simply because a spare bit is set to '1'.

The second octet of a variable length information element indicates the total length of the contents of that information element regardless of the coding of the first octet (i.e. the length is calculated starting from octet 3). It is the binary coding of the number of octets of the contents, with bit 1 as the least significant bit (2°).

An optional variable-length information element may be present, but empty (zero length). The receiver should interpret this as if that information element

Bluetooth.

was absent. Similarly, an absent information element should be interpreted by the receiver as if that information element was empty.

7.4.2 Audio control

The purpose of the Audio control information elements is to indicate information relating to the control of audio.

8	7	6	5	4	3	2	1	Octets				
0	0	0	0	0	1	1	0	1				
Lengt	Length of contents of information element (octets)											
Contr	ol informa	ation	.,,					3				

Figure 7.6:

Cont	rol i	nfoi	mati	on (d	ctet	3)		
	Bits	;						
	7	6	5	4	3	2	1	•
	0	0	0	0	0	0	0	Volume increase
	0	0	0	0	0	0	1	Volume decrease
₹	ō	Õ	Ô	0	0	1	0	Microphone gain increase
- 1	ō	Õ	a	0	0	1	1	Microphone gain decrease
- 1	ŏ	x	X	X	X	X	X	Reserved for Bluetooth standardization
ļ	1	x	X	X	X	X	X	Company specific

Table 7.5: Audio Control information element coding

7.4.3 Bearer capability

The purpose of the Bearer capability information elements is to indicate a requested or available bearer service.

If this information element is absent, the default Bearer capability is Link type Synchronous Connection-Oriented with packet type HV3, using CVSD coding for the User information layer 1.

8	7	6	5	4	3	2	1	Octets
0	0	0	0	0	0	1	0	1
Length	of conte	ents of info	ormation e	element (o	ctets)			2
Link ty	/pe							3

Figure 7.7:

Link type element coding = 00000000 (SCO)

Bluetooth.

	User information layer 1	Packet type	4
- 1			

Figure 7.8:

Link type element coding = 00000001 (ACL)

Flags	4
Service type	5
	6
Token Rate	7
	8
	9
	10
Token Bucket Size (bytes)	11
	12
	13
	14
Peak Bandwidth (bytes/second)	15
	16
	17
	18
Latency (microseconds)	19
	20
	21
	22
Delay Variation (microseconds)	23
, , , , , , , , , , , , , , , , , , ,	24
	25
User information layer 3 User information layer 2	26

Figure 7.9:

Note: the Quality of Service is repeated at TCS level, as only TCS has the knowledge of end-to-end Quality of Service requirements.

Bluetooth.

Link type (octet 3)
Bits 8 7 6 5 4 3 2 1 0 0 0 0 0 0 0 0 Synchronous Connection-Oriented 0 0 0 0 0 0 0 1 Asynchronous Connection-Less 0 0 0 0 0 0 1 0 None All other values are reserved
Octet 4 coding (Link type element coding = 000000000)
Packet type (octet 4)
Bits 5 4 3 2 1 0 0 1 0 1 HV1 0 0 1 1 0 HV2 0 0 1 1 1 HV3 0 1 0 0 DV All other values are reserved
User information layer 1 (octet 4)
Bits 8 7 6 0 0 1 CVSD 0 1 0 PCM A-law 0 1 1 PCM μ-law All other values reserved
Octets 4-26 coding (Link type element coding = 000000001)
The details of the coding Octets 4-25 can be found in L2CAP, see L2CAP, Section 6 on page 289
User information layer 2 (octet 26)
Bits 4 3 2 1 0 0 0 0 RFCOMM over L2CAP All other values are reserved
User information layer 3 (octet 26)
Bits 8 7 6 5 0 0 0 0 Not specified 0 0 0 1 PPP 0 0 1 0 IP All other values reserved
Octet 4 coding (Link type element coding = 000000010)
Octet 4 is absent

Table 7.6: Bearer capability information element coding

Bluetooth.

7.4.4 Call class

The purpose of the Call class is to indicate the basic aspects of the service requested. This element allows the user to indicate the use of default attributes, thereby reducing the length of the set-up message.

*********	8	7	6	5	4	3	2	1	Octets		
	1	1	0	0	0	0	0	0	1		
	Call Class										

Figure 7.10:

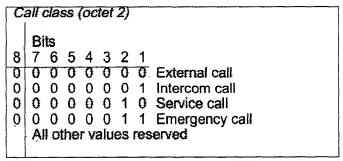


Table 7.7: Call class information element coding

Note

- An external call is a call to/from an external network; e.g. the PSTN.
- An intercom call is a call between Bluetooth devices.
- A service call is a call for configuration purposes.
- An emergency call is an external call using a dedicated emergency call number, using specific properties.

Bluetooth.

7.4.5 Called party number

The purpose of the Called party number information element is to identify the called party of a call.

8	7	6	5	4		3	1	2	1	Octets	
0	0	0	0	0		1	(0	1	1	
	Len	gth of con	tents of ir	formatio	n eleme	ent (octets))		2	
0	Ty	pe of nun	nber	N	umberi	ition	3				
0	0 Number digits (IA5 characters) (Note)										

Note – The number digits appear in multiple octet 4's in the same order in which they would be entered, that is, the number digit which would be entered first is located in the first octet 4.

Figure 7.11:

Bi	ŝ			
7	6	5		
0	0	0	Unknown	
0	0	1	International number	
0	1	0	National number	
0	1	1	Network specific number	
1		Ð		
1	1	0	Abbreviated number	
1	4	4	Reserved for extension	
			110001100 101 012101-11	
			values are reserved	
Numb	erir			
Numb Bi	erir ts	ng p	values are reserved	
Numb Bi	erir ts 3		alues are reserved an identification (octet 3) 1 0 Unknown	
Numb Bi 4	ts 3	ng p 2 0	alues are reserved an identification (octet 3) Unknown SDN/telephony numbering plan	£.16
Numb Bi 4	ts 3 0	2 0 0	ralues are reserved lan identification (octet 3) 1 0 Unknown 1 ISDN/telephony numbering plan E	:.16
Numb Bi 4 0 0 0	ts 3 0 0	2 0 0 1	alues are reserved an identification (octet 3) Unknown ISDN/telephony numbering plan E Data numbering plan Rec. X.121 Reserved	
Numb Bi 4 0	ts 3 0 0 0 1	2 0 0	alues are reserved lan identification (octet 3) 1 0 Unknown 1 ISDN/telephony numbering plan E 1 Data numbering plan Rec. X.121 0 Reserved	

Table 7.8: Called party information element coding

Bluetooth.

7.4.6 Calling party number

The purpose of the Calling party number information element is to identify the origin of a call.

8	7 6	5	4	3	2	1	Octets				
0	0 0	0	0	1	0	0	1				
	Length of contents of information element (octets)										
0	Type of nu	mber	Numb	3							
0	Presentation indicator	0	0	0		ening cator	4				
0	0 Number digits (IA5 characters)										

Figure 7.12:

Time of no	umber (octet 3)
rype or ne	umber (octet s)
0 0 0 1 0 1 1 0 1 1	5 0 Unknown 1 International number 0 National number 1 Network specific number 0 Subscriber number 0 Abbreviated number 1 Reserved for extension
All Out	let values are reserved
Numberin	ng plan identification (octet 3)
0 0 0 0 1 1 1 0 1 0 All oth	· · · · · · · · · · · · · · · · · ·
Bits 7 6 0 0 0 1 1 0 1 1 All ott	Presentation allowed Presentation restricted Number not available due to interworking Reserved her values are reserved

Table 7.9: Calling party information element coding

Bluetooth.

Scree	ning indicator (octet 4)	
Bit 2 0 0 1 1 Al	1 User-provided, not screened 1 User-provided, verified and passed 0 User-provided, verified and failed 1 Network provided other values are reserved	

Table 7.9: Calling party information element coding

7.4.7 Cause

The purpose of the Cause is to indicate the remote side of the cause of the failure of the requested service.

8	7	6	5	4	3	2	1	Octets			
1	1 .	0	0	0	0	0	1	1			
	Cause value										

Figure 7.13:

C	ause (octet 2)	harris
8	Bits 7 6 5 4 3 2 1 These 7 bits are coded alike the Cause value subfield defined in Section 2.2.5 of ITU-T Recommendation Q.850[2].	

Table 7.10: Cause information element coding

7.4.8 Clock offset

The purpose of the Clock offset information element is to indicate the Bluetooth clock offset used.

8	7	6	5	4	3	2000	2	1	Octets
0	0	0	0	0	0		0	0	1
	Leng	gth of conte	ents of info	ormation	elemen	t (octets)		2
			Clock	offset			***		3
									4

Figure 7.14:

Bluetooth.

C	lock offset coding (octe	t 3 and 4)
8	Bits (octet 3) 7 6 5 4 3 2 1 8 Contains bits 16-2 of I	Bits (octet 4) 7 6 5 4 3 2 1 Bluetooth clock

Table 7.11: Clock offset information element coding

7.4.9 Company specific

The purpose of the Company specific information element is to send non-standardized information.

8	7	0000	6		5	***************************************	4		3	***********	2	100000000	1	Octets
0	0	_ : _	0		0		0		1		1		1	1
	Le	ngth	of co	nten	ts of i	nforr	natio	n elei	ment	(octe	ets)			2
411				Coi	npan	y Ide	ntifica	ation				-		3
				Col	mpan	y Ide	ntific	ation						4
			(Comp	oany :	spec	ific co	nten	ts					,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
			a									_		L+2

Figure 7.15:

-		Bits (octet 3) 7 6 5 4 3 2 1 8								Bits (octet 4)						
8	7	6	*	4	3	2	1	8		6		•	3	2	1	
Ō	Ö	Ō	0	0	0	0	0	Œ	0	0	0	0	0	0	0	Ericsson
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	Nokia Mobile Phones
0	Ð	Ð	0	0	0	0	0	0	0	0	0	0	0	1	0	Intel Corporation
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	IBM Corporation
0	0	Ð	0	0	0	0	Đ	0	Q	ø	Ø	O	1	0	0	Toshiba Corporation

Table 7.12: Company specific information element coding

483

Bluetooth.

7.4.10 Configuration data

The purpose of the Configuration data information element is to indicate the Configuration data.

		Bluetooth	address	or word	membe	: 6 1				10
		<u></u>			<u> </u>					10
										11
	Lin	k key to b	e used to	wards W	/UG me	mbe	er 1			
								_		26
0	Ir	nternal nur	nber of W	/UG me	mber n (IA5	char	acte	er)	3+((n-1)*24)
0	ir	nternal nui	mber of W	/UG me	mber n (IA5	char	racte	er)	4+((n-1)*24)
	<u> </u>	<u> </u>							-	5+((n-1)*24)
		Bluetooth	n address	of WUG	e membe	er n				•••
										10+((n-1)*24)
										11+((n-1)*24)
		.1.1		words V	VIIC ma	mh	orn			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Link key to be used towards WUG member n										***************************************
ł										

Note – The internal number (2 digits) appears in octets 3 and 4 in the same order in which they would be entered; that is, the number digit which would be entered first is located in octet 3.

Note – The octets 3-26 are repeated for all n WUG members.

Figure 7.16:

Bluetooth.

7.4.11 Destination CID

The purpose of the Destination CID information element is to enable the remote side to associate the established L2CAP channel with the ongoing call. The Destination CID is identical to the Destination CID (DCID) exchanged in the Configuration Request packet (see L2CAP, Section 5.4 on page 280).

8	7	6	5	4		3		2	1	Octets		
0	0	0	0	0		0		1	1	1		
	Length of contents of information element (octets)											
			DCID	byte 1						3		
	DCID byte 0											

Figure 7.17:

7.4.12 Keypad facility

The purpose of the Keypad facility information element is to convey IA5 characters; e.g. entered by means of a terminal keypad.

0000000	8	7	30-0000	6	· war war	5	0.00000000	4	**********	3	***************************************	2	***************************************	1	Octets
اً	1	1	2	0		0		0		1		0		0	1
	0			Key	/pad	facilit	y info	ormat	ion (IA5 ch	arad	cter)			2

Figure 7.18:

7.4.13 Progress indicator

The purpose of the Progress indicator information element is to describe an event that has occurred during the life of a call.

********	8	7	6	5	4	3	2	1	Octets
Ì	1	1	0	0	0	0	-1	0	1
	Ō			Prog	ress descr	iption			2

Figure 7.19:

Bits
7 6 5 4 3 2 1
0 0 0 1 0 0 0 In-band information or appropriate pattern is now available

All other values reserved

Table 7.13: Progress indicator information element coding

Bluetooth.

7.4.14 SCO Handle

The purpose of the SCO handle information element is to enable the remote side to associate the established SCO link with the ongoing call. The SCO handle is identical to the SCO handle exchanged in the LMP_SCO_link_req sent by the piconet master (see LMP, Section 3.21 on page 219).

8	7	6	5	4	3	2	1	Octets			
1	1	0	0	0	1	0	1	1			
	SCO handle value										

Figure 7.20:

7.4.15 Sending complete

The purpose of the Sending complete information element is to optionally indicate completion of called party number.

4	8	7	6	5	4	3	2	1	Octet
	1	0	1	0	0	0	0	1	1

Figure 7.21:

7.4.16 Signal

The purpose of the Signal information element is to convey information to a user regarding tones and alerting signals.

8	7	6	5	4	3	2	1	Octets
1	1	0	0	0	0	1	1	1
			Signa	l value				2

Figure 7.22:

	Si	gna	va	lue	(OC	tet :	2)		
		Bit							
	8	7	6	5					
	0	1	0	0	0	0	0	0	External call
Ì	0	1	0	0	0	0	0	1	Internal call
	0	1	0	0	0	0	1	Ü	Call back Reserved for Bluetooth standardization
Ī	0	X	Х	Х	Х	Х	X	Х	Reserved for Bluetooth standardization
	1	X	χ	X	X	X	X	X	Company specific
	·								* * `

Table 7.14: Signal information element coding

Bluetooth.

8 MESSAGE ERROR HANDLING¹

8.1 PROTOCOL DISCRIMINATION ERROR

When a message is received with a protocol discriminator coded other than the ones defined in Section 7.2 on page 472, that message shall be ignored.

8.2 MESSAGE TOO SHORT OR UNRECOGNIZED

When a message is received that is too short to contain a complete message type information element, that message shall be ignored.

When a message is received that contains a complete message type information element, but with a value which is not recognized as a defined message type, that message shall be ignored.

8.3 MESSAGE TYPE OR MESSAGE SEQUENCE ERRORS

Whenever an unexpected message, except RELEASE or RELEASE COM-PLETE message is received in any state other than the Null state, that message shall be ignored.

When an unexpected RELEASE message is received, the receiving side shall disconnect and release the bearer channel if established, return a RELEASE COMPLETE message, stop all timers, and enter the Null state.

When an unexpected RELEASE COMPLETE message is received, the receiving side shall disconnect and release the bearer channel if established, stop all timers, and enter the Null state.

8.4 INFORMATION ELEMENT ERRORS

The information elements in a message shall appear (if present for information elements indicated as optional) in the exact order as indicated in Section 6.

When a message is received which misses a mandatory information element, or which contains a mandatory information element with invalid content, the message shall be ignored.

In case the error occurred with a mandatory information element in a SETUP message, a RELEASE COMPLETE message shall be returned, either with cause #96, mandatory information element is missing, or with cause #100, invalid information element contents.

^{1.} In this section, when it is stated to ignore a certain message or part of a message (information element), this shall be interpreted as to do nothing – as if the (part of the) message had never been received.

Bluetooth.

When a message is received which has an unrecognized information element, or has an optional information element with an invalid content, or has a recognized information element not defined to be contained in that message, the receiving side shall ignore the information element.

Information elements with a length exceeding the maximum length (as given in Section 7 on page 471) shall be treated as an information element with invalid content.

Bluetooth.

9 PROTOCOL PARAMETERS

9.1 PROTOCOL TIMERS

Timer name	Value
T301	Minimum 3 minutes
T302	15 seconds
T303	20 seconds
T304	30 seconds
T305	30 seconds
T308	4 seconds
T310	30 –120 seconds
T313	4 seconds
T401	8 second
T402	8 seconds
T403	4 second
T404	2.5 seconds
T405	2 seconds
T406	20 seconds

Table 9.1: Timer values

Bluetooth.

10 REFERENCES

- [1] Q.931, "Digital Subscriber Signalling System No. 1(DSS 1) ISDN User-Network interface Layer 3 Specification for Basic Call Control", 03/93
- [2] Q.850, "Digital Subscriber Signalling System No. 1 General Usage of cause of location in the Digital Subscriber Signalling system No. 1 and the signalling system No. 7 ISDN User Part", 03/93

Bluetooth.

11 LIST OF FIGURES

Figure 1.1:	TCS within the Bluetooth stack435	
Figure 1.2:	Point-to-point signalling in a single-point configuration436	
Figure 1.3:	Signalling in a multi-point configuration436	
Figure 1.4:	TCS Architecture437	
Figure 2.1:	Call establishment message flow445	
Figure 2.2:	Call clearing message flow448	
Figure 3.1:	Obtain access rights message flow451	
Figure 3.2:	Configuration distribution message flow452	
Figure 3.3:	Fast inter-member access message flow454	
Figure 4.1:	Connectionless TCS message flow455	
Figure 5.1:	Calling line identity message flow456	
Figure 5.2:	DTMF start & stop message flow457	
Figure 7.1:	Protocol discriminator472	
Figure 7.2:	Message type472	
Figure 7.3:	Single octet information element format	
Figure 7.4:	Double octet information element format474	
Figure 7.5:	Variable length information element format474	
Figure 7.6:	476	i
Figure 7.7:	476	į
Figure 7.8:	477	,
Figure 7.9:	477	
Figure 7.10:	479	,
Figure 7.11:	480	;
Figure 7.12:	481	1
Figure 7.13:	482	<u>-</u>
Figure 7.14:	482	<u>-</u> د
Figure 7.15:	483	3 A
Figure 7.16:	482	ŧ =
Figure 7.17:	101) =
Figure 7.18:	401	
Figure 7.19:	48))
Figure 7.20:	40	Э С
Figure 7.21:	10	2
Figure 7.22:	48	U
	JI TOO State Diagram	3
Figure A: Fu	ıll TCS State Diagram49 ean TCS State Diagram49	4
Figure 8: Le	ean ICS State Diagram	•

Bluetooth.

12 LIST OF TABLES

Table 6.1:	ALERTING message content	460
Table 6.2:	CALL PROCEEDING message content	460
Table 6.3:	CONNECT message content	
Table 6.4:	CONNECT ACKNOWLEDGE message content	461
Table 6.5:	DISCONNECT message content	462
Table 6.6:	INFORMATION message content	462
Table 6.7:	PROGRESS message content	463
Table 6.8:	RELEASE message content	463
Table 6.9:	RELEASE COMPLETE message content	464
Table 6.10:	SETUP message content	
Table 6.11:	SETUP ACKNOWLEDGE message content	
Table 6.12:	Start DTMF message content	465
Table 6.13:	Start DTMF Acknowledge message content	
Table 6.14:	Start DTMF Reject message content	466
Table 6.15:	Stop DTMF message content	466
Table 6.16:	Stop DTMF Acknowledge message content	467
Table 6.17:	ACCESS RIGHTS REQUEST message content	
Table 6.18:	ACCESS RIGHTS ACCEPT message content	467
Table 6.19:	ACCESS RIGHTS REJECT message content	
Table 6.20:	INFO SUGGEST message content	
Table 6.21:	INFO ACCEPT message content	
Table 6.22:	LISTEN REQUEST message content	469
Table 6.23:	LISTEN SUGGEST message content	469
Table 6.24:	LISTEN ACCEPT message content	
Table 6.25:	LISTEN REJECT message content	
Table 6.26:	CL INFO message content	
Table 7.1:	General message format	
Table 7.2:	Protocol discriminator	
Table 7.3:	Message type	472
Table 7.4:	Information element identifier coding	474
Table 7.5:	Audio Control information element coding	
Table 7.6:	Bearer capability information element coding	478
Table 7.7:	Call class information element coding	479
Table 7.8:	Called party information element coding	
Table 7.9:	Calling party information element coding	
Table 7.10:	Cause information element coding	
Table 7.11:	Clock offset information element coding	
Table 7.12:	Company specific information element coding	
Table 7.13:	Progress indicator information element coding	
Table 7.14:	Signal information element coding	
Table 9.1:	Timer values	

Bluetooth.

APPENDIX 1 - TCS CALL STATES

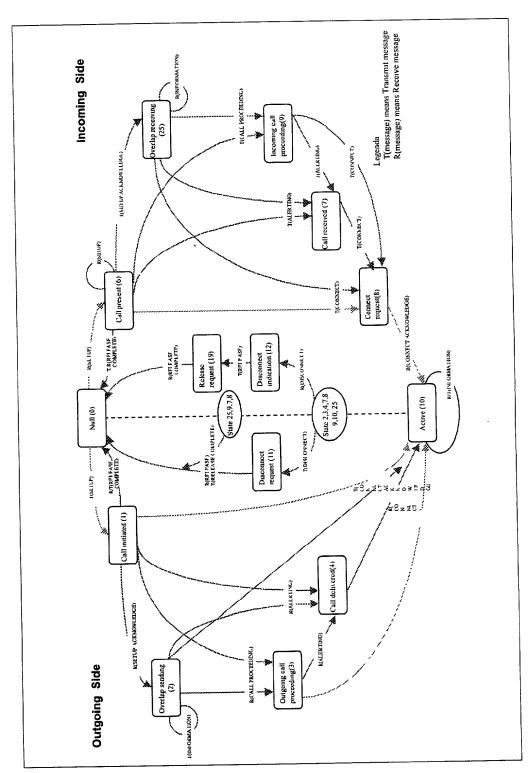


Figure A: Full TCS State Diagram

Bluetooth.

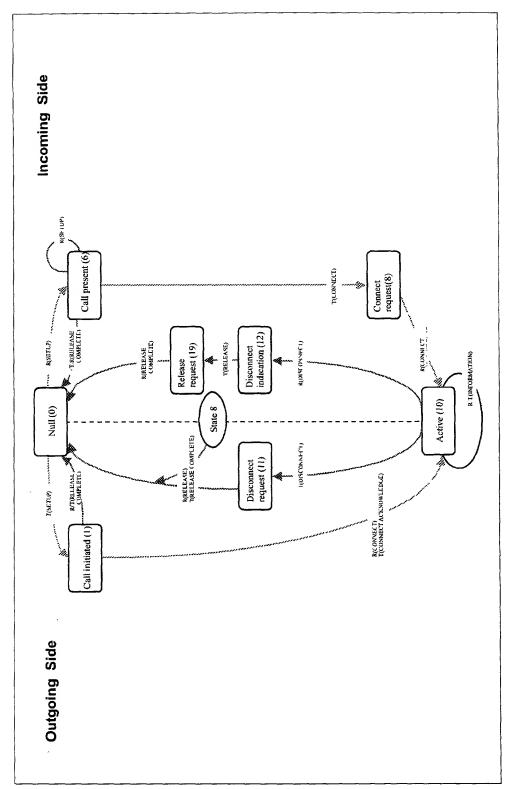


Figure B: Lean TCS State Diagram

Part F:4

INTEROPERABILITY REQUIREMENTS FOR BLUETOOTH AS A WAP BEARER

PPP Adaptation

The first that

4.3

Many of the characteristics of Bluetooth devices are shared with the target platforms for the Wireless Application Protocol. In some cases, the same device may be enabled for both types of communication. This document describes the interoperability requirements for using Bluetooth with PPP as the communications bearer for WAP protocols and applications.

Bluetooth.

Bluetooth.

CONTENTS

1	Intro	duction		499			
	1.1	Docum	nent Scope	499			
2	The	The Use of WAP In the Bluetooth Environment50					
	2.1	Value-	added Services	500			
	2.2	Usage	Cases				
		2.2.1	Briefcase Trick	500			
		2.2.2	Forbidden Message	501			
		2.2.3	WAP Smart Kiosk	501			
3	WAP	Service	es Overview	502			
	3.1	WAP E	Entities	502			
		3.1.1	WAP Client	502			
		3.1.2	WAP Proxy/Gateway	503			
		3.1.3	WAP Server	503			
	3.2	WAP F	Protocols	503			
		3.2.1	Wireless Datagram Protocol (WDP)	504			
		3.2.2	Wireless Transaction Protocol (WTP)	504			
		3.2.3	Wireless Transport Layer Security (WTLS)	504			
		3.2.4	Wireless Session Protocol (WSP)	504			
	3.3		asting WAP and Internet Protocols				
		3.3.1	UDP/WDP	504			
		3.3.2	WTP/TCP	505			
		3.3.3	WTLS/SSL	505			
		3.3.4	WSP/HTTP	505			
		3.3.5	WML/HTML	505			
		3.3.6	WMLScript/JavaScript	505			

Interd	perability	Requirem	ents for Bi	luetooth as a WAP Bearer	Bluetooth.
4	WAP	in the B	luetooth	Piconet	506
	4.1	WAP Se 4.1.1		mmunications by the Client Device	
		4.1.2	4.1.1.1 Termina	Discovery of Servicestion by the Client Device	
		4.1.3	Initiation	by the Server Device	507
	4.2	Implem 4.2.1	entation	Discovery of Servicesof WAP for Bluetoothanagement Entity	508
		4.2.2		Asynchronous Notifications Alternate Bearers	508
	4.3	Network 4.3.1		t for WAP	
5	Inter	operabili	ity Requ	irements	511
	5.1 5.2	_		Interoperability	
6	Serv	ce Disc	overy	*******************************	512
	6.1 6.2 6.3	SDP Se	ervice Re rotocol D	ecordsata Unitsry procedure	512 514
7	Refe	rences			515

Bluetooth.

1 INTRODUCTION

1.1 DOCUMENT SCOPE

This document is intended for Bluetooth implementers who wish to take advantage of the dynamic, ad-hoc characteristics of the Bluetooth environment in providing access to value-added services using the WAP environment and protocols.

Bluetooth provides the physical medium and link control for communications between WAP client and server. This document describes how PPP may be used to achieve this communication.

The information contained in this document is not sufficient to allow the implementation of a general-purpose WAP client or server device. Instead, this document provides the following information:

- An overview of the use of WAP in the Bluetooth environment will explain
 how the concept of value-added services fits within the Bluetooth vision.
 Examples are given of how the WAP value-added services model can be
 used to fulfil specific Bluetooth usage models.
- The WAP Services Overview attempts to place the WAP environment in a familiar context. Each component of WAP is introduced, and is contrasted with equivalent Internet protocols (where applicable).
- A discussion of WAP in the Bluetooth Piconet describes how the particular structure of Bluetooth communications relates to WAP behaviors.
- Finally, the Interoperability Requirements describe the specific Bluetooth features that must be implemented in order to ensure interoperability between any two WAP enabled Bluetooth devices.

Bluetooth.

2 THE USE OF WAP IN THE BLUETOOTH ENVIRONMENT

2.1 VALUE-ADDED SERVICES

The presence of communications capabilities in a device is unlikely to be an end in itself. The end users are generally not as interested in the technology as in what the technology allows them to do.

Traditional telecommunications relies on voice communications as the single application of the technology, and this approach has been successful in the marketplace. As data communications services have become more widely available, there is increasing pressure to provide services that take advantage of those data capabilities.

The Wireless Application Protocol Forum was formed to create a standardsbased framework, in which value-added data services can be deployed, ensuring some degree of interoperability.

2.2 USAGE CASES

The unique quality of Bluetooth, for the purposes of delivering value-added services, is the limited range of the communications link. Devices that incorporate Bluetooth are ideally suited for the receipt of location-dependent services. The following are examples of how the WAP client / server model can be applied to Bluetooth usage cases.

2.2.1 Briefcase Trick

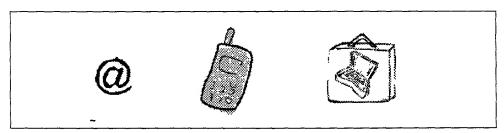


Figure 2.1: The 'Briefcase Trick' Hidden Computing Scenario

The Briefcase Trick usage case allows the user's laptop and mobile phone to communicate, without user intervention, in order to update the user's e-mail. The user can review the received messages from the handset, all without removing the laptop from its storage in a briefcase.

2.2.2 Forbidden Message

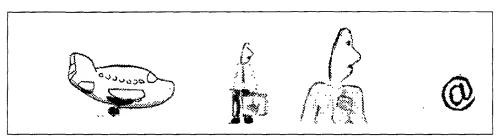


Figure 2.2: The 'Forbidden Message' Hidden Computing Scenario

The Forbidden Message usage case is similar to the briefcase trick. The user can compose messages in an environment where no dial-up connection is possible. At a later time the laptop wakes up, and checks the mobile phone to see if it is possible to send the pending messages. If the communications link is present, then the mail is transmitted.

2.2.3 WAP Smart Kiosk

The WAP Smart Kiosk usage case allows a user to connect a mobile PC or handheld device to communicate with a kiosk in a public location. The kiosk can provide information to the device that is specific to the user's location. For example, information on flights and gates in an airport, store locations in a shopping centre, or train schedules or destination information on a railway platform.

Bluetooth.

3 WAP SERVICES OVERVIEW

The Wireless Application Protocol is designed to provide Internet and Internet-like access to devices that are constrained in one or more ways. Limited communications bandwidth, memory, processing power, display capabilities and input devices are all factors driving the development of WAP. Although some devices may only exhibit some of the above constraints, WAP can still provide substantial benefit for those devices as well.

The WAP environment typically consists of three types of device: the WAP Client device, the WAP Proxy/gateway and WAP Server. In some cases the WAP Proxy/gateway may also include the server functionality.

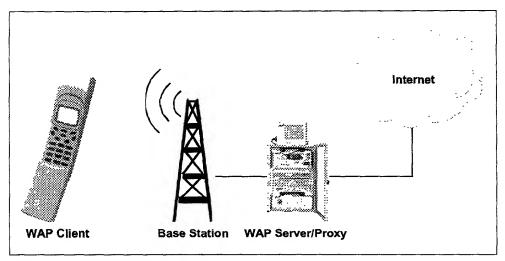


Figure 3.1: Typical WAP Environment

3.1 WAP ENTITIES

3.1.1 WAP Client

The WAP Client device is usually found in the hands of the end user. This device can be as powerful as a portable computer, or as compact as a mobile phone. The essential feature of the client is the presence of some type of display and some type of input device.

The WAP Client is typically connected to a WAP Proxy/gateway through a wireless network. (Figure 3.2 on page 503) This network may be based on any available technology. The WAP protocols allow the network to exhibit low reliability and high latency without interruption in service.

Bluetooth.

3.1.2 WAP Proxy/Gateway

The WAP Proxy/gateway acts as an interface between the wireless network, and the larger Internet. The primary functions of the proxy are to provide DNS name resolution services to WAP client devices and translation of Internet protocols and content formats to their WAP equivalents.

3.1.3 WAP Server

The WAP Server performs a function that is similar to a server in the Internet world. In fact, the WAP server is often an HTTP server. The server exists as a storage location for information that the user can access. This 'content' may include text, graphics, and even scripts that allow the client device to perform processing on behalf of the server.

The WAP Server logic may exist on the same physical device as the Proxy/gateway, or it may reside anywhere in the network that is reachable from the Proxy/gateway.

The server may fill the role of an HTTP server, a WSP server, or both.

3.2 WAP PROTOCOLS

The WAP environment consists of a layered protocol stack that is used to isolate the user agents from the details of the communications network. Figure 4.1 on page 506 illustrates the general architecture of the WAP protocol stack. Bluetooth will provide an additional data bearer service, appearing at the bottom of this diagram.

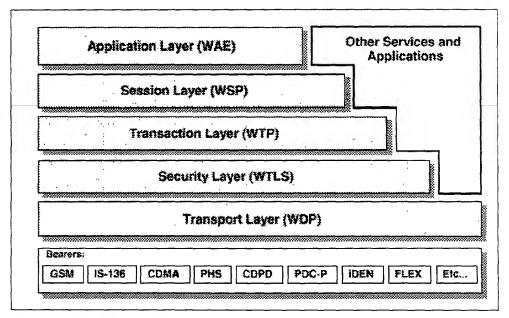


Figure 3.2: WAP Protocol Stack

Bluetooth.

3.2.1 Wireless Datagram Protocol (WDP)

The WDP layer provides a service interface that behaves as a socket-based UDP implementation. For a bearer service based on IP, then this layer is UDP. For bearer which do not provide a UDP service interface, then an implementation of WDP must be provided to act as an adaptation layer to allow socket-based UDP datagrams over the native bearer.

3.2.2 Wireless Transaction Protocol (WTP)

The WTP layer provides a reliable datagram service on top of the WDP (UDP) layer below.

3.2.3 Wireless Transport Layer Security (WTLS)

The WTLS layer is an optional component of the protocol stack that provides a secure data pipe between a client WSP session and its peer server WSP session. In the current version of the WAP specification, this session will terminate at the WAP server. There is currently a proposal before the WAP Forum for a proxy protocol, which will allow the intermediate WAP proxy to pass WTLS traffic across the proxy/gateway without decrypting the data stream.

3.2.4 Wireless Session Protocol (WSP)

The WSP layer establishes a relationship between the client application, and the WAP server. This session is relatively long-lived and able to survive service interruptions. The WSP uses the services of the WTP for reliable transport to the destination proxy/gateway.

3.3 CONTRASTING WAP AND INTERNET PROTOCOLS

The intent and implementation of the WAP protocol stack has many parallels with those of the Internet Engineering Task Force (IETF). The primary objective of the WAP Forum has been to make Internet content available to devices that are constrained in ways that make Internet protocols unsuitable for deployment.

This section compares the roles of the WAP protocol stack's layers with those of the IETF.

3.3.1 UDP/WDP

At the most basic layer, WAP and Internet protocols are the same. The WAP stack uses the model of a socket-based datagram (UDP) service as its transport interface.

Some Internet protocols also use the UDP service, but most actually use a connection-oriented stream protocol (TCP).

Bluetooth.

3.3.2 WTP/TCP

The wireless transport protocol (WTP) provides services that, in some respects, fill the same requirements as TCP. The Internet Transmission Control Protocol (TCP) provides a reliable, connection-oriented, character-stream protocol that is based on IP services. In contrast, WTP provides both reliable and unreliable, one-way and reliable two-way message transports. The transport is optimized for WAP's 'short request, long response' dialogue characteristic. WTP also provides message concatenation to reduce the number of messages transferred.

3.3.3 WTLS/SSL

The Wireless Transport Layer Security (WTLS) is derived from the Secure Sockets Layer (SSL) specification. As such, it performs the same authentication and encryption services as SSL.

3.3.4 WSP/HTTP

Session services in WAP are provided by the Wireless Session Protocol (WSP). This protocol incorporates the semantics and functionality of HTTP 1.1, while adding support for long-lived sessions, data push, suspend and resume. Additionally, the protocol uses compact encoding methods to adapt to narrowband communications channels.

3.3.5 WML/HTML

The markup language used by WAP is a compact implementation that is similar to HTML, but optimized for use in hand-held devices. WML is an XML-defined markup language.

3.3.6 WMLScript/JavaScript

WAP also incorporates a scripting language that is similar to JavaScript, but adapted to the types of constrained devices that WAP is targeted for.

Bluetooth.

4 WAP IN THE BLUETOOTH PICONET

In many ways, Bluetooth can be used like other wireless networks with regard to WAP. Bluetooth can be used to provide a bearer for transporting data between the WAP Client and its adjacent WAP Server.

Additionally, Bluetooth's *ad hoc* nature provides capabilities that are exploited uniquely by the WAP protocols.

4.1 WAP SERVER COMMUNICATIONS

The traditional form of WAP communications involves a client device that communicates with a Server/Proxy device using the WAP protocols. In this case the Bluetooth medium is expected to provide a bearer service as specified by the WAP architecture.

4.1.1 Initiation by the Client Device

When a WAP client is actively 'listening' for available Bluetooth devices, it can discover the presence of a WAP server using Bluetooth's Service Discovery Protocol.

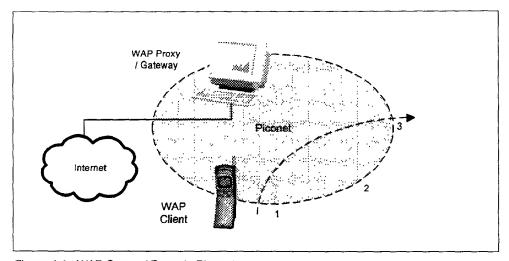


Figure 4.1: WAP Server / Proxy in Piconet

In Figure 4.1, stage 1 the WAP Client device is moving into range of the WAP Proxy/gateway's piconet. When the client detects the presence of the WAP proxy/gateway, it can automatically, or at the client's request, connect to the server.

Bluetooth.

4.1.1.1 Discovery of Services

The client must be able to determine the specific nature of the WAP proxy/ gateway that it has detected. It is expected that the Bluetooth Service Discovery Protocol will be used to learn the following information about the server:

- · Server Name this is a user readable descriptive name for the server.
- Server Home Page Document Name this is the home page URL for the server. This is optional.
- Server/Proxy Capability indicates if the device is a WAP content server, a
 Proxy or both. If the device is a Proxy, it must be able to resolve URLs that
 are not local to the Server/Proxy device.

In Figure 4.1, stage 2, the device is communicating with the WAP proxy/gateway. All WAP data services normally available are possible.

4.1.2 Termination by the Client Device

In Figure 4.1, stage 3, the device is exiting the piconet. When the device detects that communication has been lost with the WAP proxy/gateway, it may optionally decide to resume communications using the information obtained at discovery.

For example, a client device that supports alternate bearers may query the alternate address information of the server when that capability is indicated. The information should be cached for later access because the client device may leave the piconet at any time, and that information will no longer be available.

In the WAP Smart Kiosk example above, if the user wishes to continue receiving information while out of Bluetooth range, the Kiosk would provide an Internet address to the client device. When Bluetooth communications are not possible, the device could use cellular packet data to resume the client-server session.

This capability is implementation-dependent, and is provided here for illustrative purposes only.

4.1.3 Initiation by the Server Device

An alternative method of initiating communications between a client and server is for the server to periodically check for available client devices. When the server device discovers a client that indicates that it has WAP Client capability, the server may optionally connect and push data to the client.

The client device has the option of ignoring pushed data at the end user's discretion.

Bluetooth.

4.1.3.1 Discovery of Services

Through the Bluetooth Service Discovery Protocol, the server can determine the following information about the client:

- Client Name this is a friendly format name that describes the client device
- Client capabilities this information allows the server to determine basic information regarding the client's Bluetooth-specific capabilities

4.2 IMPLEMENTATION OF WAP FOR BLUETOOTH

In order to effectively implement support for WAP over Bluetooth, certain capabilities must be considered.

4.2.1 WDP Management Entity

Associated with an instance of the WDP layer in the WAP Protocol Stack is an entity that is responsible for managing the services provided by that layer. The WDP Management Entity (WDP-ME) acts as an out-of-band mechanism for controlling the protocol stack.

4.2.1.1 Asynchronous Notifications

The WDP-ME will need to be able to generate asynchronous notifications to the application layer when certain events occur. Example notifications are:

- · New Client Node Detected
- New Server Node Detected
- Client Node Signal Lost
- Server Node Signal Lost
- Server Push Detected (detected as unsolicited content)

Platform support for these events is implementation-specific. All of the listed events may be derived through the Bluetooth Host Controller Interface (page 517), with the exception of Server Push.

4.2.1.2 Alternate Bearers

An implementation of WAP on a particular device may choose to support multiple bearers. Methods of performing bearer selection are beyond the scope of this document. The procedure to be followed is implementation-dependent. See Section 4.1.2 above.

Bluetooth.

4.2.2 Addressing

Two basic types of addressing are being used in the WAP environment: User Addressing and Proxy/gateway Addressing. User addressing describes the location of objects within the network, and is independent of the underlying bearer. Proxy/Gateway Addressing describes the location of the WAP proxy/gateway that the device is communicating with. Proxy/Gateway addressing is dependent on the bearer type.

The end user deals mainly with Uniform Resource Locators (URL). These addresses are text strings that describe the document that is being accessed. Typically, the Proxy/gateway in conjunction with Internet Domain Name.

Servers resolve these strings into network addresses.

The address of the WAP Proxy/gateway is usually a static value that is configured by the user or network operator. When the user enters a URL, the request is forwarded to the configured WAP proxy/gateway. If the URL is within the domain of a co-located server, then it indicates that the document is actually WAP content. If the URL is outside of the WAP proxy/gateway's domain, then the WAP Proxy/gateway typically uses DNS name resolution to determine the IP address of the server on which the document resides.

The client device would first identify a proxy/gateway that is reachable through Bluetooth, then it would use the service discovery protocol to present the user with a server name or description. When the user selects a server, then the WAP client downloads the home page of the server (as determined by the discovery process; see section 4.1.1.1 on page 507) Once the user has navigated to the home page of the desired server, then all subsequent URLs are relative to this home page. This scenario presumes that the WAP Proxy/gateway and WAP Content server are all co-located in the Bluetooth device, although this structure is not required for interoperability.

A WAP Proxy/gateway/Server will typically provide a default URL containing the home page content for the server. A proxy-only device typically provides no URL or associated content.

4.3 NETWORK SUPPORT FOR WAP

The following specifies a protocol stack, which may be used below the WAP components. Support for other protocol stack configurations is optional, and must be indicated through the Bluetooth Service Discovery Protocol.

4.3.1 PPP/RFCOMM

Devices that support Bluetooth as a bearer for WAP services using PPP provide the following protocol stack support:

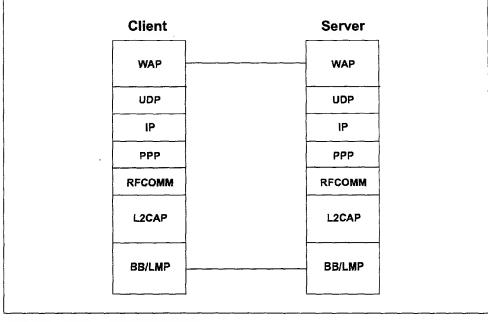


Figure 4.2: Protocol Support for WAP

For the purposes of interoperability, this document assumes that a WAP client conforms to the role of Data Terminal as defined in LAN Access Profile using PPP [6]. Additionally, the WAP server or proxy device is assumed to conform to the role of the LAN Access Point defined in [6].

The Baseband (page 33), LMP (page 185) and L2CAP (page 245) are the OSI layer 1 and 2 Bluetooth protocols. RFCOMM (page 385) is the Bluetooth adaptation of GSM TS 07.10 [1]. SDP (page 323) is the Bluetooth Service Discovery Protocol.

PPP is the IETF Point-to-Point Protocol [3]. WAP is the Wireless Application Protocol stack and application environment [5].

1,7

-4

Bluetooth.

5 INTEROPERABILITY REQUIREMENTS

5.1 STAGE 1 - BASIC INTEROPERABILITY

Stage 1 interoperability for WAP over Bluetooth (all mandatory):

- Provide WAP Class A device compliance [7]
- · Provide, through service discovery mechanisms, the network address for devices that support WAP proxy/gateway functionality.

5.2 STAGE 2 - ADVANCED INTEROPERABILITY

Stage 2 interoperability for WAP over Bluetooth (mandatory):

- All Stage 1 interoperability requirements are supported
- Provide Server Name and information about Server/Proxy capabilities through service discovery.
- · Provide Client Name and information about Client Capabilities through service discovery.
- · Asynchronous Notifications for Server.
- Asynchronous Notifications for Client.

Bluetooth.

6 SERVICE DISCOVERY

6.1 SDP SERVICE RECORDS

Service records are provided as a mechanism through which WAP client devices and proxy/gateways become aware of each other dynamically. This usage differs from other WAP bearers in that the relationship between the two devices will be transitory. That is, a Bluetooth device will not have a bearer-specific address configured or provisioned to a specific proxy/gateway.

Clients and proxy/gateways become aware of each other as they come in proximity of one another. The Bluetooth Service Discovery Protocol allows the devices to query the capabilities of each other as listed in the Interoperability Requirements section of this document.

Table 6.1 shows the service record for the WAP Proxy/gateway device.

lŧ	em	Definition	Туре	Value	AttriD	Req
s	erviceClassIDList				0x0001	М
S	erviceClass0	WAP Proxy/Gateway	UUID	WAP		M
-	luetoothProfile escriptorList					м
	ProfileDescriptor0				0x0009	M
	Profile	Supported Profile	UUID	LANAccess UsingPPP [4]		м
	Version	Profile Version	Uint16	(varies)		М
1	rotocol escriptorList					0
	Descriptor0	UDP	UUID	UDP	,	0
	Parameter0	WSP Connectionless Session Port No.	Uint16	9200 (default)		0
	Parameter1	WTP Session Port No.	Uint16	9201 (default)		0
	Parameter2	WSP Secure Connectionless Port No.	Uint16	9202 (default)		0
	Parameter3	WTP Secure Session Port No.	Uint16	9203 (default)	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	O
	Parameter4	WAP vCard Port No.	Uint16	9204 (default)		0
	Parameter5	WAP vCal Port No.	Uint16	9205 (default)		0
	Parameter6	WAP vCard Secure Port No.	Uint16	9206 (default)		0
	Parameter7	WAP vCal Secure Port No.	Uint16	9207 (default)		O

Table 6.1: Service Record format for WAP Proxy/Gateway devices

Bluetooth.

ltem	Definition	Type	Value	AttriD	Req
ServiceName	Displayable Text name	String	(varies, e.g. 'Airport infor- mation')		
NetworkAddress	IP Network Address of Server	Uint32	(varies)		M
			0x01 = Origin Server;		
WAPGateway*	Indicates if device is	Uint8	0x02 = Proxy;	М	М
The Galoria	origin server or proxy		0x03 = Origin Server and Proxy		
HomePageURL	URL of home page document	URL			C1 [†]

Table 6.1: Service Record format for WAP Proxy/Gateway devices

- *. Stage 2 interoperability requirements.
- †. If this parameter is omitted, then a default is assumed for origin servers as: http://networkaddress/index.wml

it	em	Definition	Туре	Value	AttriD	Req
s	erviceClassIDList				0x0001	М
ServiceClass0		WAP Client	UUID	WAP_CLIENT	;	M
1 -	luetoothProfile escriptorList					M
	ProfileDescriptor0	-			0x0009	M
	Profile	Supported Profile	מוטט	LANAccess UsingPPP [4]		М
	Version	Profile Version	Uint16	(varies)		M
S	erviceName	Displayable Text name of client	String	(varies)		0

Table 6.2: Service Record format for WAP Client devices

Bluetooth.

6.2 SDP PROTOCOL DATA UNITS

Table 6.3 shows the specified SDP PDUs (Protocol Data Units), which are required for WAP Interoperability.

2011		Ability to	o Send	Ability to Retrieve	
PDU No.	SDP PDU	WAP Client	WAP Proxy	WAP Client	WAP Proxy
1	SdpErrorResponse	M	M	M	M
2	SdpServiceSearchAttributeRequest	М	0	М	М
3	SdpServiceSearchAttributeResponse	M	М	M	M

Table 6.3: SDP PDU:s

6.3 SERVICE DISCOVERY PROCEDURE

In the simplest form, the signaling can be like this:

WAP Client or Proxy		WAP Client or Proxy
	SdpServiceSearchAttributeRequest	
	SdpServiceSearchAttributeResponse	
	<=====================================	

WAP service discovery procedures are symmetrical. Each device must be able to handle all of the PDUs without regard for the current device role. A minimal implementation must return the service name string.

Bluetooth.

7 REFERENCES

- [1] TS 101 369 (GSM 07.10) version 6.2.0
- [2] Simpson, W., Editor, "The Point-to-Point Protocol (PPP)", STD 50, RFC 1661, Daydreamer, July 1994.
- [3] Simpson, W., Editor, "PPP in HDLC Framing", STD 51, RFC 1662, Day-dreamer, July 1994.
- [4] See Appendix VIII, "Bluetooth Assigned Numbers" on page 1009
- [5] Wireless Application Protocol Forum, "Wireless Application Protocol", version 1.0, 1998
- [6] Bluetooth Special Interest Group, "Bluetooth LAN Access Profile using PPP"
- [7] Wireless Application Protocol Forum, "WAP Conformance", Draft version 27 May 1998

Windows Wireless Architecture

Tim Moore Lead Program Manager Windows Networking Microsoft Corporation

Agenda

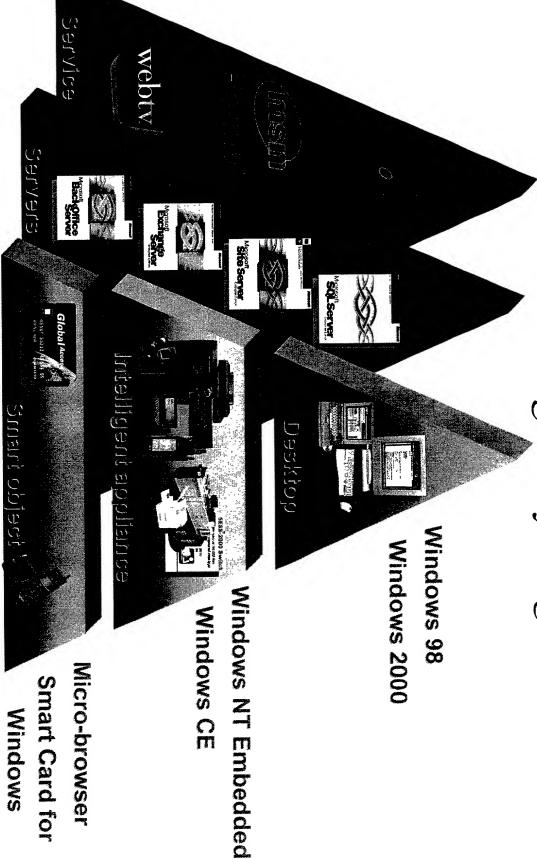
- Wireless trends
- WAN, LAN, PAN
- Scenarios
- Adhoc, home, small business
- Enterprise, ISP
- Wireless architecture
- Summary
- Call to action
- More information

Wireless Trends

- IP networks
- Always connected
- Increased bandwidth
- Convenience
- Moving from vertical market to horizontal markets
- Moving from proprietary to standards based
- Proliferation of smart devices
 New scenarios enabled
- Outsourcing
- Adhoc networks

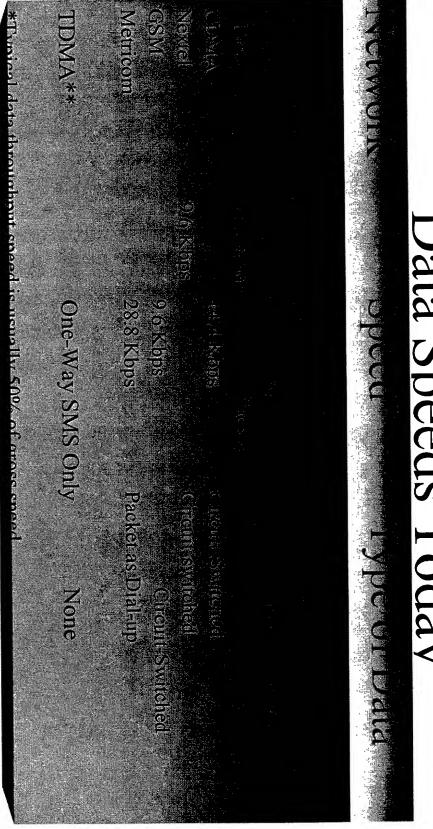
Information Anytime, Anywhere

Connecting Everything



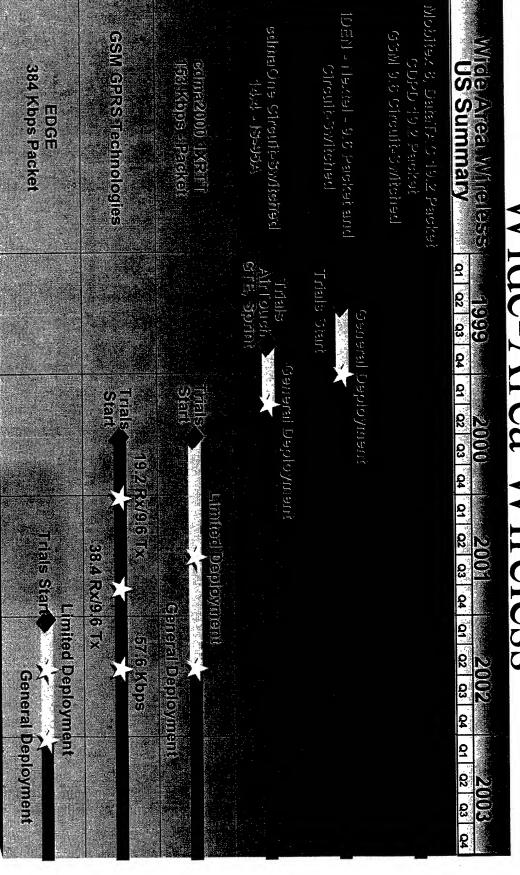
THE REPORT OF THE REAL PROPERTY AND THE REAL

Data Speeds Today



**TDMA systems do not support data in the U.S. at this time

Wide-Area Wireless

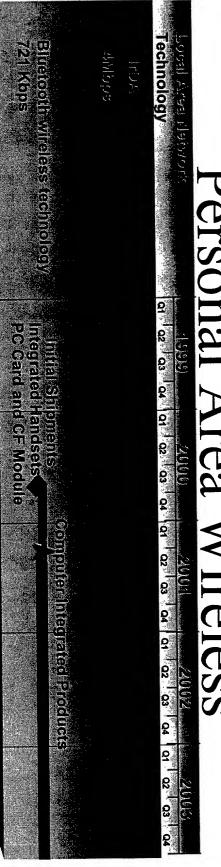


Materials from Andrew Seybold-Microsoft Exchange Conference 1999

Local-Area Wireless

P802.11a.5.GHz 54 Mbps Direct Sequence Spread Spectrum	P802 (1)5 (DSSS) 2.4 GHz 11 Mbps Direct Sequence Spread Spectrum	802.11 (D333) 2.4 OHz 1 or 2 Mbos Direct Sequence Spread Spectrum Hiperian 20,6 Maps 20,6 Maps 110jh Pariomange Fadio LAN	802.14 (FHSS) 2.4 GHz 1 Mbps Freq. Hopped Spread Specialin	Losal Arsa Nsbyork Technology
Specifications Approved	initial.			(1999) 24000 Car Car
Shipments				(959) 2006 2007 2008 2008 2008 2002 2006 2006 2008 2008

Personal Area Wireless



Personal Area Wireless

- IrDA
- Around since 1994
- Available on every PC and lots of devices
- >20 million existing IrDA devices
- Camera, PDAs, cellphones, printers, keyboards
- Exploding market fueled by Bluetooth momentum
- Bluetooth wireless technology is a defacto standard
- Proliferation of smart devices, convenience of cable replacement, and new usage scenarios

Scenarios

- Adhoc
- Home
- Small business Enterprise
- ISP



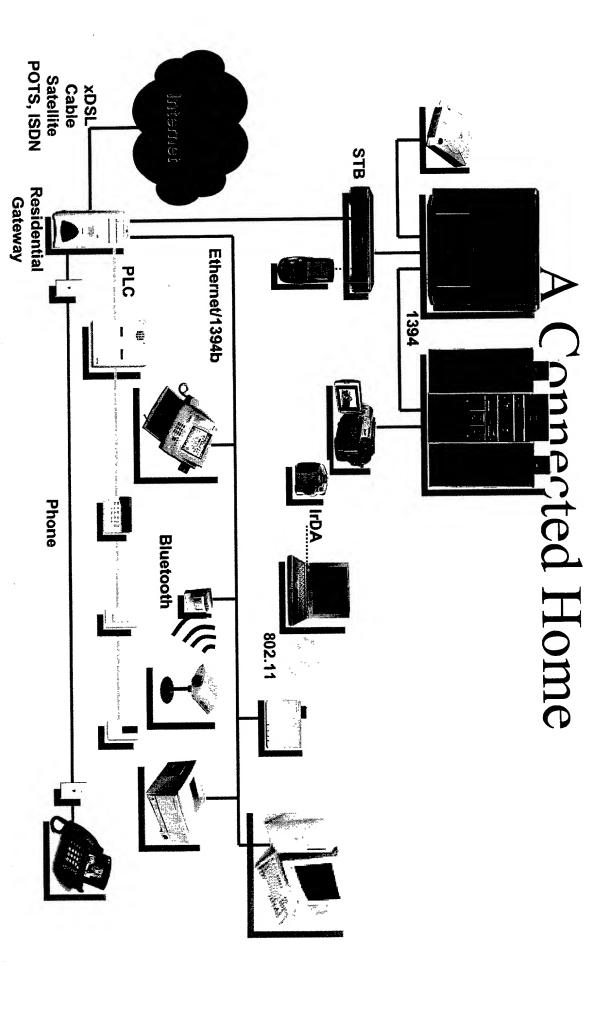
connected Many diverse devices to be



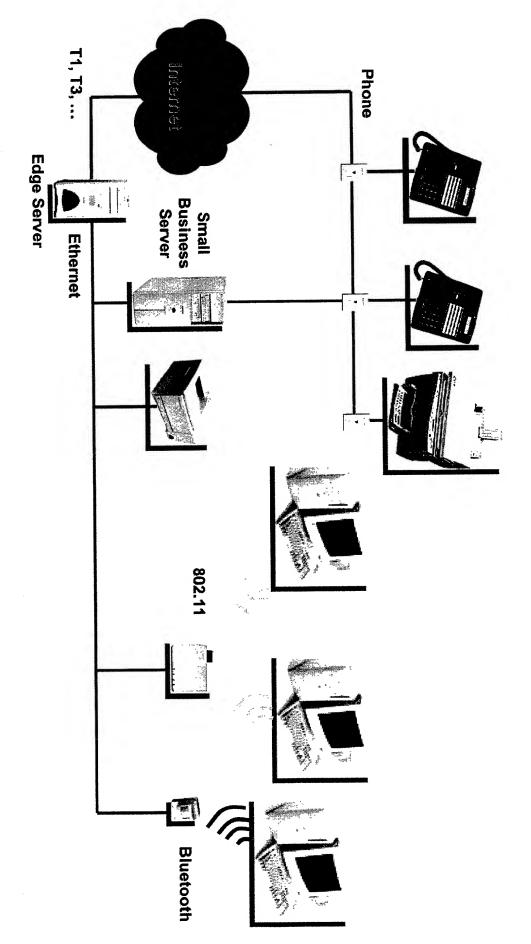


Phones, Pagers PC companions handheld PCs Books, tablets,

The train that the train that the train that the



A Connected Small Office



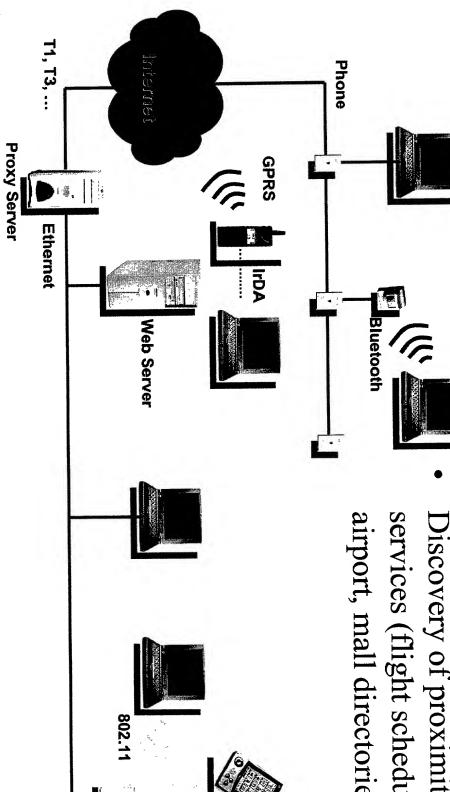
Proxy Server

Enterprise

- End-user can access the enterprise wireless network transparently over a secure connection
- access to the enterprise wireless LAN The network administrator has control over which users have
- the enterprise outsource their authentication to Enterprise can offer its employees access via ISPs which
- End-user has IP connectivity as soon as a CDPD or a GPRS modem is plugged in
- Make cellphones an always connected Internet access point using
- connected cellphone out of range of LAN can continue to conference via IP End-User can use Netmeeting with wireless LAN, when

An ISP Connected Public

Space



services (flight schedules at airport, mall directories, ...) Discovery of proximity

ISP

- Need mixed technologies
- Higher speed in hot spots, e.g., 802.11
- Need authentication so ISPs can charge
- Allow ISPs to integrate into existing Radius systems
- Allows ISP roaming agreements
- Same as outsource dial
- Need to be able to provision unauthenticated users

Wireless Architecture

- "Just works"
- Always connected
- Unified transport: IP
- Mobility
- Unified security model
- Adhoc
- QoS
- Performance

Wireless Architecture



Location

Newyork

ស្នាវវេធ១ 4545

APIS APIS Offil-Ital()

njehvork

Networking

APIs

ទីលាមេនាវាន

(Direct)

DHCP

Networking Services

NetBT

iP packet ព្រក្រទីព្រ

Jepanska.

GMP

Packet scheduler

Protocol stacks

NDIS 5.1





Affected by Wireless

Just Works

- No configuration
- Especially when roaming
- CDPD
- Configure Network Equipment Identifier
- Configure network name and security keys Per location

802.11

- Bluetooth wireless technology
- Configure PIN numbers
- Per device

802.11 Configuration

- Current 802.11 networks need to be configured with name of the network
- Roaming between multiple networks difficult 1s implemented especially when security
- Automatically find a wireless network
- If Access point is beaconing network name, attempt to use that network
- If no infrastructure available then switch to adhoc mode

Always Connected

- Permanent IP connectivity should not use dial-up model
- A CDPD card should appears as a LAN card
- A GPRS, EDGE or 3G card or cellphone should appear as a LAN card
- GPRS Terminal Type Recommendations
- Cellphone needs to be Type A (voice and packet)
- PC-Card can be Type C (packet only)
- Implement an NDIS driver or use **Kemote NDIS**
- Remote NDIS over Bluetooth connections

Remote NDIS

- devices that provide Remote NDIS enables a bus-agnostic connection to network access
- command language Remote NDIS is both a driver architecture and a



Unified Transport: IP

- All other media except Bluetooth wireless technology support always connected IP
- Ethernet over point-to-point Bluetooth connections
- L2 bridge gives an adhoc L2 network
- Adhoc applications use UPnP over IP
- Expect large numbers of wireless connected devices
- Move to IPv6 for addresses

Mobility

- Applications should not rely on having a network available all the time
- Network connection can disappear at anytime
- Applications should reconnect automatically if the network appears
- Clients hold state about the network
- IP address
- Routes
- Networks hold state about the client
- Multicast distribution
- Quality of service
- Secure access
- Machine name to IP address mapping
- How to detect when this state is out of date
- Applications also hold state about the network
- TCP connections
- E.g. Proxies, firewalls, etc.

Mobility

- Detect roaming
- Mediasense detects working/non-working interfaces
- Mediasense detects interfaces changing their network connection
- IP address
- Mediasense triggers a DHCP renew; If renew fails, DHCP gets a new IP address
- DHCP updates DNS when an address changes
- TCP/IP removes IP addresses if NIC not connected
- Mobile IP allows IP address to stay the same when roaming

Mobile IP

- Mobile IP keeps the application IP address the same
- IPv4 has two options
- Change the network interface address to a local IP address
- Use an ARP proxy to keep the same IP address
- IPv6 only has first option
- Mobile IP Issues
- How to route efficiently
- IPv6 fixes this issue
- Firewall traversal
- Time to get a local address
- Doesn't allow Voice over IP roaming
- applications Doesn't address any of the other issues with multicast, QoS, security,
- GPRS and 3G have network layer mobility
- No plans to support Mobile IP until IPv6

Mobility

- Multicast
- Mediasense triggers IGMP refresh on roaming
- QoS
- Mediasense triggers RSVP refresh on roaming
- Routes
- Mediasense triggers router detect (IRDP) on roaming
- Default interface metrics should depend interface speed
- Routes to no longer existing interface addresses are removed
- Security
- Mediasense triggers network authentication refresh
- **Applications**
- Need to retry connections on connection failure and mediasense
- Configurations based on network location

Network Location API

- Network location is a hint to the application of the network the machine is connected to
- Accessible via Winsock API
- Query for the connected networks
- WSALookupServiceBegin
- WSALookupServiceNext
- WSALookupServiceEnd
- Request for notification when the connected networks changes
- WSANSIoctl (,SIO_NSP_NOTIFY_CHANGE,...)
- use this API Applications that need configuration per network should
- E.g., application proxies

Security

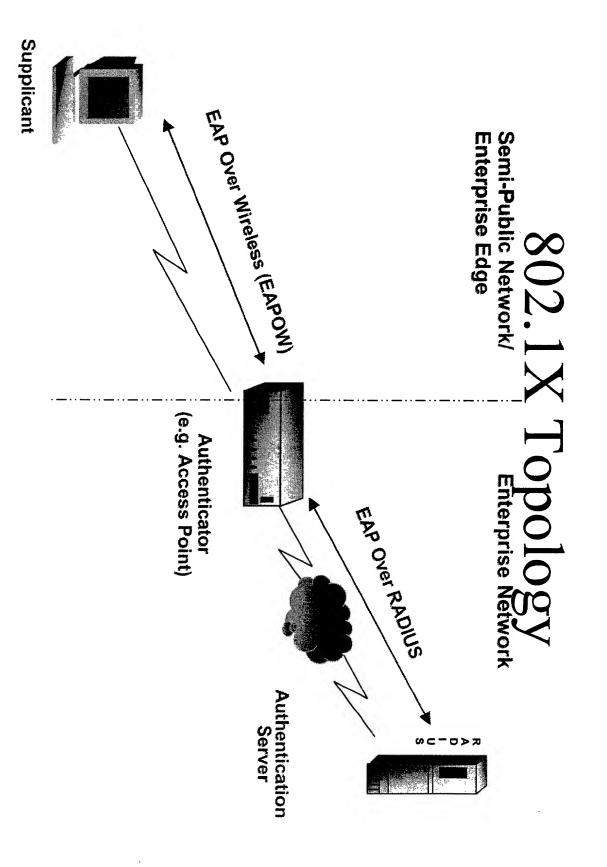
- Secure access to resources in the network
- This is Windows login
- Secure transfer of data over the network
- This is IPSec
- Integrated into Windows credentials using PKI and Kerberos
- Secure access to the network
- This is available for RAS and VPNs
- Integrated into Windows credentials using PKI (EAP) and Radius
- Supports roaming of identities
- No secure access to LAN networks
- Very important for Wireless

Wireless Security Issues

- User loses wireless NIC, doesn't report it
- Without user authentication, Intranet now accessible by attackers
- Without centralized accounting and auditing, no means to detect unusual activity
- Users who don't log on for periods of time
- Users who transfer too much data, stay on too long
- Multiple simultaneous logins
- Logins from the "wrong" machine account
- With global keys, large scale re-keying required

Wireless Deployment Issues

- User administration
- Integration with existing user administration tools required (RADIUS, LDAP-based directories)
- Create a Windows group for wireless
- access Any user or machine who is a member of the group has wireless
- address identification Identification via User-Name easier to administer than MAC
- Usage accounting and auditing desirable
- Key management
- Static keys difficult to manage on clients, access points
- Proprietary key management solutions require separate user databases



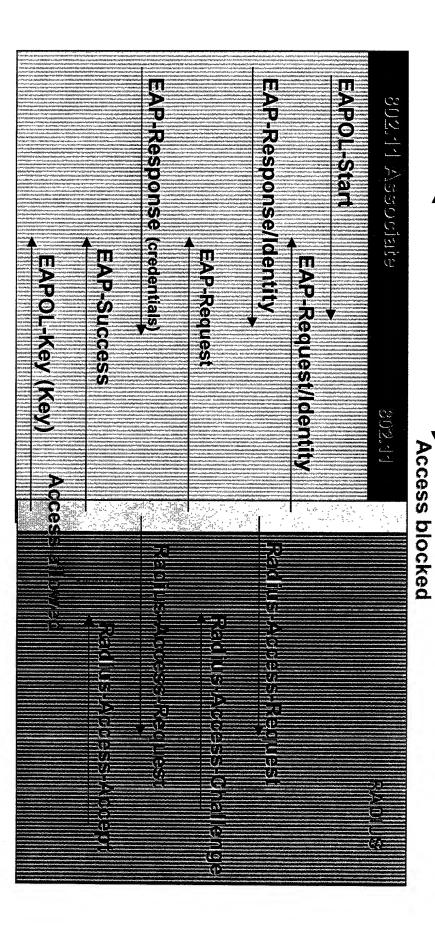
${ m IEEE}~802.1{ m X}$

- management Enables interoperable user identification, centralized authentication, key
- Leverages existing standards: EAP, RADIUS
- places Compatible with existing roaming technologies, enabling use in hotels and public
- User-based identification
- Identification based on Network Access Identifier (RFC 2486) enables support for roaming access in public spaces (RFC 2607)
- Dynamic key management
- Centralized user administration
- Support for RADIUS (RFC 2138, 2139) enables centralized authentication, authorization and accounting
- within RADIUS RADIUS/EAP (draft-ietf-radius-ext-07.txt) enables encapsulation of EAP packets
- Supported on Ethernet, Token Ring and 802.11

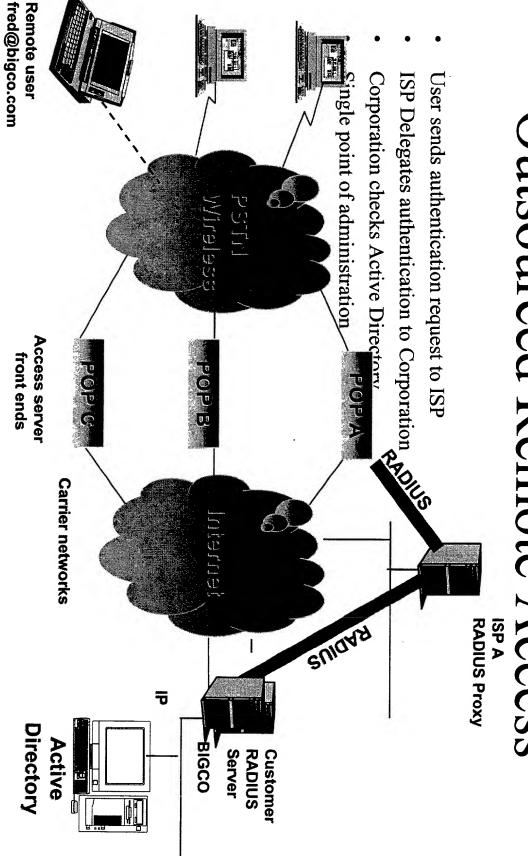
Extensible Authentication **Protocol**

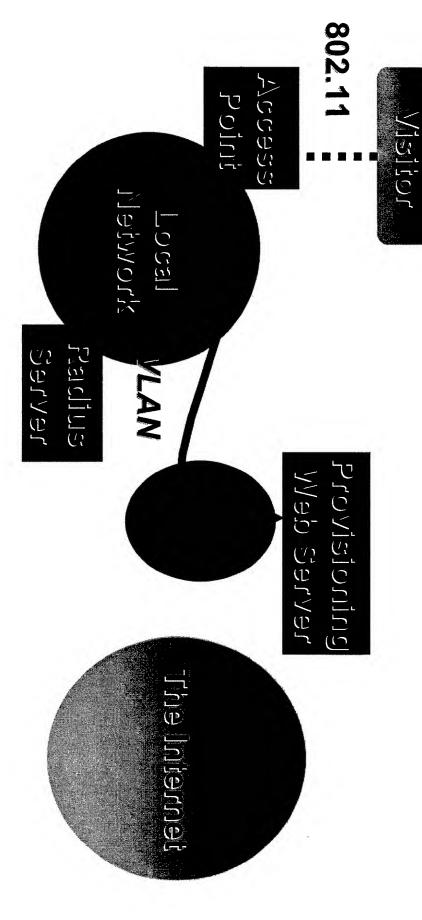
- Used by PPP for RAS and VPN
- Allows support for a number of authentication mechanisms
- EAP designed to allow additional authentication methods to be deployed with no changes to the access point or client NIC
- RFC 2284 includes support for password authentication (EAP-MD5), One-Time Passwords (OTP)
- Security Dynamics Windows 2000 supports smartcard authentication (RFC 2716) and
- Radius server used for authentication and authorization Integrated into Active DirectoryTM users and groups
- Supports cross authentication for roaming





Radius Server





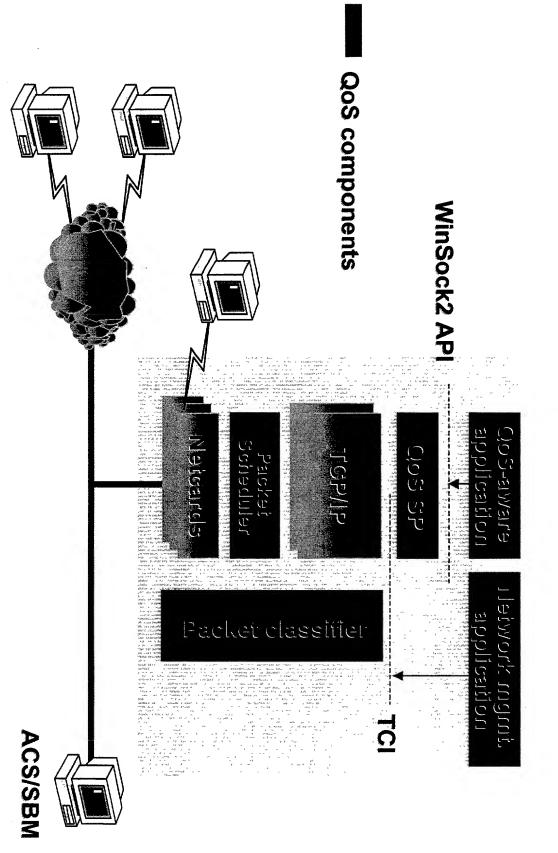
Bluetooth Security

- To connect to a Bluetooth device requires its PIN
- PIN is per device not per service
- Great for personal single function devices
- E.g., protect cellphone from being dialed
- Problem for adhoc devices/applications
- Require PIN for each device
- Obtain access to all services on device
- Need security at a higher level and no PIN
- Adhoc FTP user intervention required so why need a pin?
- roaming PANs Adhoc PAN do not want a PIN otherwise cannot setup
- Business card exchange should be push to a destination

GPRS Security

- GPRS uses GSM Authentication
- Authentication is between the mobile station and the network
- Need authentication between PC and the Bluetooth mobile station
- Bluetooth PIN

Microsoft® QoS Components

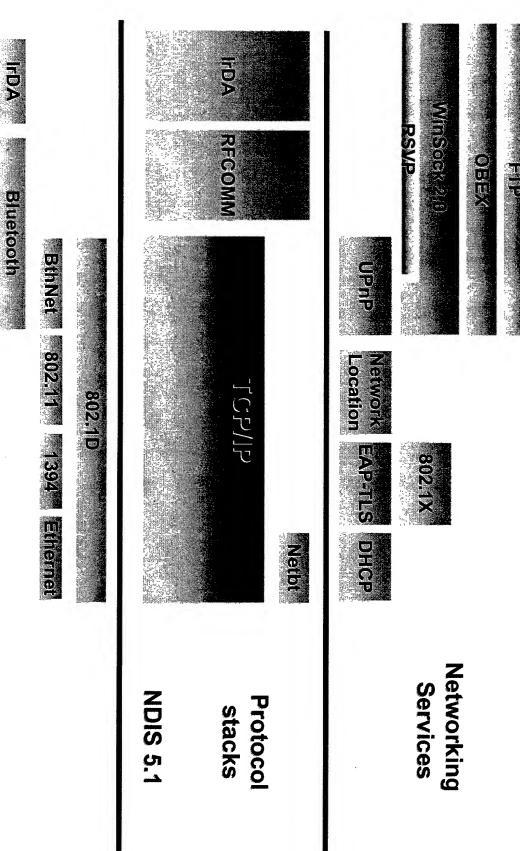


Carte and a man is not in the fact that they are the fact that the fact that

802.11 QoS

- 802.1p support
- Priority tagging of Ethernet frames
- 802.11 NIC driver
- Use NDIS priority field to prioritize access from client to wireless network
- Add 802.1p header for wired network
- to client based on 802.1p Access point prioritizes access from wired network
- Subnetwork bandwidth manager in access point tor admission control

Adhoc Architecture



No Network Infrastructure

- Address assignment
- APIPA when no DHCP server
- ICS contains DHCP server for adhoc home network
- Name Resolution
- NetBT broadcast for adhoc name resolution
- ICS contains DNS proxy and DDNS support for the adhoc home network
- Service Discovery Protocols
- SSDP protocol enables UPnP discovery
- SDP protocol enables Bluetooth wireless technology discovery
- IrLAP protocol enables IrDA discovery

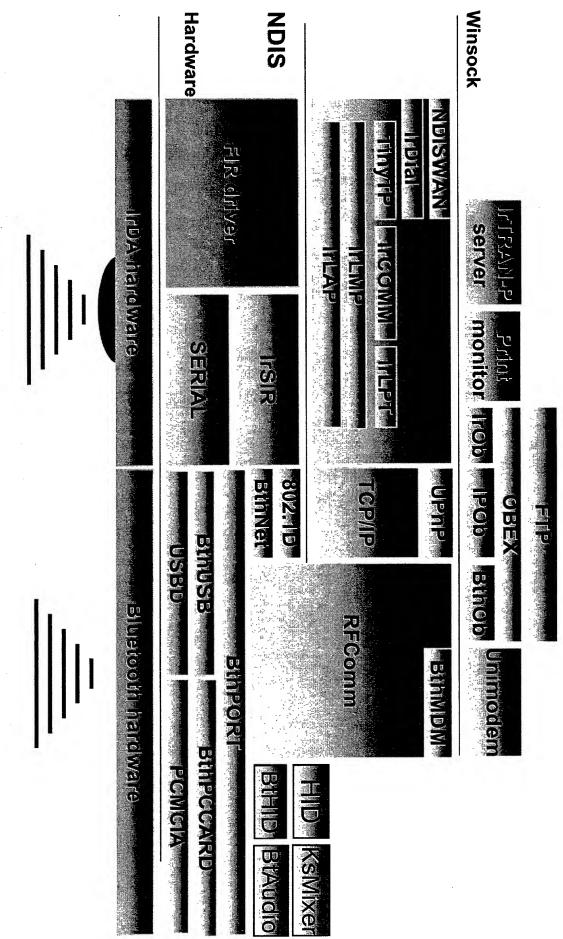
Temporary Networks

- Wireless allows for networks to be setup easily
- Interconnections not organized
- Multiple interconnections to destinations
- Loops in the network
- L2 Spanning tree
- Self organizing networks
- Removes loops

Ad Hoc Ethernet Networks

- Ethernet hubs
- Ethernet cross-over cables
- 1394
- Host to Host USB cables
- 802.11 can form adhoc mode
- Automatically switch to adhoc mode when no access points in range
- Bluetooth wireless technology
- IrDA

IrDA/Bluetooth Architecture



IrDA Applications

- File transfer
- Integrated into shell
- Image exchange from camera
- Dial-up networking via cellphone
- Printing
- Synchronization
- ActiveSync®

Bluetooth Applications

- Subset of IrDA
- File transfer
- Integrated into IrDA ftp transfer
- Dial-up Networking via cellphone
- particular media IR and Bluetooth applications are tied to
- Do not inter-operate

- Ad Hoc Applications
 UPnP is the integration point for ad hoc applications
- UPnP applications and services are available over any IP network
- Ethernet, Wireless LAN, 1394, etc.

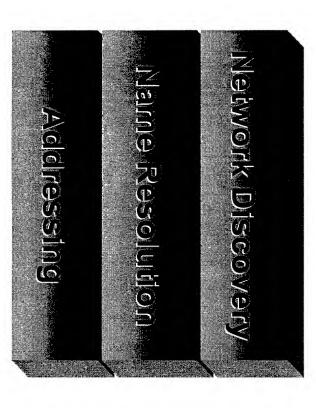
PnP Architecture Reference

Usage

Description



- Standardized protocols
- Standardized XML descriptions
- Simple discovery
- Locate devices/services on-the-fly
- Standards-based



How It Works



Usage phase





Negotiation phase



Discovery phase



Wireless Performance

- TCP has many features optimized for wireless in Windows 2000
- Improved RTT estimate
- Improved window sizes
- Fast retransmit
- Select acknowledgement
- Acknowledge packets
- Improved time-out initiation
- Very important for wireless lossesCannot be used over the serial port
- Use Remote NDIS
- Over USB, IEEE 1394, Bluetooth wireless technology

WAP

- on long thin links WAP was designed to remove some issues with TCP
- Remove 3 way handshake
- Proposals to add data on the SYN and SYN-ACK
- Reduces DOS protection
- Remove IP layer for some media
- Not removed for GPRS
- Data compression GPRS supports TCP/IP header and user data compression
- user data compression Recommend GPRS systems support protocol header and
- WML is for small screens

Summary – Wireless Is Here

- Bandwidth is growing
- Always connected wireless
- Enables new scenarios
- Driving new applications
- Security a major issue with wireless
- 802.1X allows integration into Windows user security system
- UPnP is the framework for adhoc applications

Call To Action

- Mobility
- Mediasense is required for roaming support
- Any wireless device must generate mediasense
- Implement 802.1X in network edge devices
- Switches, access points, etc.
- Adhoc services and applications

 Implement using UPnP
- Do not limit your applications to a particular wireless media

For More Information

- Bluetooth wireless technology
- IrDA -
- UPnP
- 802.11
- QoS whitepaper
- Security whitepaper
- NIC requirements whitepaper

For More Information

- RNDIS
- WinHec driver talk
- TCP/IP
- Whitepaper

For More Information

- IEEE 802.1X -RADIUS
- | | |

EAP

, '

1 Windows Wireless Architecture

Tim Moore Lead Program Manager Windows Networking Microsoft Corporation

- 2 Agenda
 - · Wireless trends
 - · WAN, LAN, PAN
 - Scenarios
 - · Adhoc, home, small business
 - · Enterprise, ISP
 - · Wireless architecture
 - Summary
 - · Call to action
 - · More information
- 3 3 Wireless Trends
- - IP networks
 - · Always connected
 - · Increased bandwidth
 - Convenience
 - Moving from vertical market to horizontal markets
 - · Moving from proprietary to standards based
 - · Proliferation of smart devices
 - · New scenarios enabled
 - Outsourcing
 - · Adhoc networks
- 4 <a> Information Anytime, Anywhere Connecting Everything
- 5 Data Speeds Today

Type of Data Network Speed*

American Mobile ARDIS 19.2/4.8 Kbps Packet **Packet** BellSouth Wireless Data 8 Kbps

Circuit-Switched 9.6/4.8 Kbps Cellular (Analog)

Packet CDPD 19.2 Kbps

Circuit-Switched 14.4 Kbps **CDMA** Circuit-switched

Nextel 9.6 Kbps Circuit-Switched 9.6 Kbps **GSM**

Packet as Dial-up 28.8 Kbps Metricom

One-Way SMS Only None TDMA**

*Typical data throughput speed is usually 50% of gross speed **TDMA systems do not support data in the U.S. at this time 6 3 Wide-Area Wireless 7 D Local-Area Wireless 8 <a> Personal Area Wireless 9 Personal Area Wireless IrDA · Around since 1994 · Available on every PC and lots of devices >20 million existing IrDA devices Camera, PDAs, cellphones, printers, keyboards · Exploding market fueled by Bluetooth momentum · Bluetooth wireless technology is a defacto standard · Proliferation of smart devices, convenience of cable replacement, and new usage scenarios 10 Scenarios Adhoc Home Small business Enterprise · ISP 11 🔊 Ad Hoc Networks 12 A Connected Home 13 A Connected Small Office 14 🔊 Enterprise · Information at your fingertips · At meetings, in the office, on the road · Reliable, secure, multimedia LAN 15 Enterprise End-user can access the enterprise wireless network transparently over a secure connection The network administrator has control over which users have access to the

Enterprise can offer its employees access via ISPs which outsource their

End-user has IP connectivity as soon as a CDPD or a GPRS modem is

Make cellphones an always connected Internet access point using GPRS

enterprise wireless LAN

authentication to the enterprise

2

- End-User can use Netmeeting with wireless LAN, when out of range of LAN can continue to conference via IP connected cellphone
- 16 An ISP Connected Public Space
 - Discovery of proximity services (flight schedules at airport, mall directories, ...)
- 17 🗀 ISP
 - · Need mixed technologies
 - Higher speed in hot spots, e.g., 802.11
 - · Need authentication so ISPs can charge
 - Allow ISPs to integrate into existing Radius systems
 - · Allows ISP roaming agreements
 - · Same as outsource dial
 - · Need to be able to provision unauthenticated users
- 18 Wireless Architecture
 - · "Just works"
 - Always connected
 - Unified transport: IP
 - Mobility
 - Unified security model
 - Adhoc
 - QoS
 - Performance
- 19 Wireless Architecture
- 20 Just Works
 - · No configuration
 - · Especially when roaming
 - CDPD
 - Configure Network Equipment Identifier
 - 802.11
 - · Configure network name and security keys
 - Per location
 - Bluetooth wireless technology
 - Configure PIN numbers
 - · Per device
- 21 **3** 802.11 Configuration
 - Current 802.11 networks need to be configured with name of the network
 - Roaming between multiple networks difficult especially when security is implemented
 - · Automatically find a wireless network
 - If Access point is beaconing network name, attempt to use that network
 - · If no infrastructure available then switch to adhoc mode
- 22 <a>Always Connected
 - Permanent IP connectivity should not use dial-up model

- · A CDPD card should appears as a LAN card
- A GPRS, EDGE or 3G card or cellphone should appear as a LAN card
 - · GPRS Terminal Type Recommendations
 - Cellphone needs to be Type A (voice and packet)
 - PC-Card can be Type C (packet only)
- Implement an NDIS driver or use Remote NDIS
 - · Remote NDIS over Bluetooth connections
- 23 🔊 Remote NDIS
 - Remote NDIS enables a bus-agnostic connection to devices that provide network access
 - · Remote NDIS is both a driver architecture and a command language
- 24 Tunified Transport: IP
 - All other media except Bluetooth wireless technology support always connected IP
 - Ethernet over point-to-point Bluetooth connections
 - · L2 bridge gives an adhoc L2 network
 - · Adhoc applications use UPnP over IP
 - · Expect large numbers of wireless connected devices
 - · Move to IPv6 for addresses
- 25 Mobility
 - Applications should not rely on having a network available all the time
 - Network connection can disappear at anytime
 - Applications should reconnect automatically if the network appears
 - · Clients hold state about the network
 - · IP address
 - Routes
 - · Networks hold state about the client
 - · Multicast distribution
 - · Quality of service
 - Secure access
 - Machine name to IP address mapping
 - · How to detect when this state is out of date
 - Applications also hold state about the network
 - TCP connections
 - · E.g. Proxies, firewalls, etc.
- 26 Mobility
 - · Detect roaming
 - Mediasense detects working/non-working interfaces

- Mediasense detects interfaces changing their network connection
- IP address
 - Mediasense triggers a DHCP renew; If renew fails, DHCP gets a new IP address
 - · DHCP updates DNS when an address changes
 - · TCP/IP removes IP addresses if NIC not connected
 - Mobile IP allows IP address to stay the same when roaming
- 27 Mobile IP
 - · Mobile IP keeps the application IP address the same
 - · IPv4 has two options
 - · Change the network interface address to a local IP address
 - · Use an ARP proxy to keep the same IP address
 - · IPv6 only has first option
 - · Mobile IP Issues
 - · How to route efficiently
 - · IPv6 fixes this issue
 - · Firewall traversal
 - · Time to get a local address
 - · Doesn't allow Voice over IP roaming
 - · Doesn't address any of the other issues with multicast, QoS, security, applications
 - GPRS and 3G have network layer mobility
 - No plans to support Mobile IP until IPv6
- 28 🗿 Mobility
 - Multicast
 - Mediasense triggers IGMP refresh on roaming
 - QoS
 - · Mediasense triggers RSVP refresh on roaming
 - Routes
 - · Mediasense triggers router detect (IRDP) on roaming
 - · Default interface metrics should depend interface speed
 - · Routes to no longer existing interface addresses are removed
 - Security
 - · Mediasense triggers network authentication refresh
 - Applications
 - · Need to retry connections on connection failure and mediasense
 - · Configurations based on network location
- 29 Network Location API
 - Network location is a hint to the application of the network the machine is connected to
 - Accessible via Winsock API

- · Query for the connected networks
 - WSALookupServiceBegin
 - WSALookupServiceNext
 - WSALookupServiceEnd
- Request for notification when the connected networks changes
 - WSANSlocti (,SIO_NSP_NOTIFY_CHANGE,...)
- · Applications that need configuration per network should use this API
 - · E.g., application proxies

30 Security

- · Secure access to resources in the network
 - · This is Windows login
- · Secure transfer of data over the network
 - · This is IPSec
 - · Integrated into Windows credentials using PKI and Kerberos
- · Secure access to the network
 - · This is available for RAS and VPNs
 - Integrated into Windows credentials using PKI (EAP) and Radius
 - · Supports roaming of identities
- · No secure access to LAN networks
 - · Very important for Wireless

31 Wireless Security Issues

- · User loses wireless NIC, doesn't report it
 - · Without user authentication, Intranet now accessible by attackers
 - Without centralized accounting and auditing, no means to detect unusual activity
 - Users who don't log on for periods of time
 - · Users who transfer too much data, stay on too long
 - Multiple simultaneous logins
 - · Logins from the "wrong" machine account
 - · With global keys, large scale re-keying required

32 Wireless Deployment Issues

- · User administration
 - Integration with existing user administration tools required (RADIUS, LDAPbased directories)
 - · Create a Windows group for wireless
 - · Any user or machine who is a member of the group has wireless access
 - Identification via User-Name easier to administer than MAC address identification
 - · Usage accounting and auditing desirable
- Key management

- · Static keys difficult to manage on clients, access points
- · Proprietary key management solutions require separate user databases
- 33 **a** 802.1X Topology
- 34 3 IEEE 802.1X
 - Enables interoperable user identification, centralized authentication, key management
 - · Leverages existing standards: EAP, RADIUS
 - Compatible with existing roaming technologies, enabling use in hotels and public places
 - · User-based identification
 - Identification based on Network Access Identifier (RFC 2486) enables support for roaming access in public spaces (RFC 2607)
 - · Dynamic key management
 - · Centralized user administration
 - Support for RADIUS (RFC 2138, 2139) enables centralized authentication, authorization and accounting
 - RADIUS/EAP (draft-ietf-radius-ext-07.txt) enables encapsulation of EAP packets within RADIUS
 - Supported on Ethernet, Token Ring and 802.11
- 35 Extensible Authentication Protocol
 - Used by PPP for RAS and VPN
 - Allows support for a number of authentication mechanisms
 - EAP designed to allow additional authentication methods to be deployed with no changes to the access point or client NIC
 - RFC 2284 includes support for password authentication (EAP-MD5), One-Time Passwords (OTP)
 - Windows 2000 supports smartcard authentication (RFC 2716) and Security Dynamics
 - · Radius server used for authentication and authorization
 - Integrated into Active Directory[™] users and groups
 - · Supports cross authentication for roaming
- 36 3 802.1X On 802.11
- 37 Outsourced Remote Access
 - · User sends authentication request to ISP
 - · ISP Delegates authentication to Corporation
 - · Corporation checks Active Directory
 - · Single point of administration
- 38 Provisioning Public Internet
- 39 🔊 Bluetooth Security
 - · To connect to a Bluetooth device requires its PIN
 - · PIN is per device not per service
 - · Great for personal single function devices
 - · E.g., protect cellphone from being dialed

- Problem for adhoc devices/applications
 - Require PIN for each device
 - · Obtain access to all services on device
- · Need security at a higher level and no PIN
 - · Adhoc FTP user intervention required so why need a pin?
 - Adhoc PAN do not want a PIN otherwise cannot setup roaming PANs
 - · Business card exchange should be push to a destination
- 40 S GPRS Security
 - · GPRS uses GSM Authentication
 - · Authentication is between the mobile station and the network
 - · Need authentication between PC and the Bluetooth mobile station
 - Bluetooth PIN
- 42 3 802.11 QoS
 - 802.1p support
 - · Priority tagging of Ethernet frames
 - 802.11 NIC driver
 - · Use NDIS priority field to prioritize access from client to wireless network
 - Add 802.1p header for wired network
 - Access point prioritizes access from wired network to client based on 802.1p
 - · Subnetwork bandwidth manager in access point for admission control
- 43 🖪 Adhoc Architecture
- 44 <a> No Network Infrastructure
 - · Address assignment
 - APIPA when no DHCP server
 - · ICS contains DHCP server for adhoc home network
 - Name Resolution
 - · NetBT broadcast for adhoc name resolution
 - · ICS contains DNS proxy and DDNS support for the adhoc home network
 - · Service Discovery Protocols
 - SSDP protocol enables UPnP discovery
 - SDP protocol enables Bluetooth wireless technology discovery
 - IrLAP protocol enables IrDA discovery
- 45 Temporary Networks
 - · Wireless allows for networks to be setup easily
 - · Interconnections not organized
 - Multiple interconnections to destinations
 - · Loops in the network
 - L2 Spanning tree
 - · Self organizing networks

- · Removes loops
- 46 Ad Hoc Ethernet Networks
 - · Ethernet hubs
 - · Ethernet cross-over cables
 - 1394
 - · Host to Host USB cables
 - · 802.11 can form adhoc mode
 - · Automatically switch to adhoc mode when no access points in range
 - · Bluetooth wireless technology
 - IrDA
- 47 <a>IrDA/Bluetooth Architecture
- 48 🗟 IrDA Applications
 - · File transfer
 - · Integrated into shell
 - · Image exchange from camera
 - · Dial-up networking via cellphone
 - Printing
 - Synchronization
 - ActiveSync®
- 49 <a> Bluetooth Applications
 - · Subset of IrDA
 - File transfer
 - · Integrated into IrDA ftp transfer
 - · Dial-up Networking via cellphone
 - IR and Bluetooth applications are tied to particular media
 - · Do not inter-operate
- 50 (a) Ad Hoc Applications
 - UPnP is the integration point for ad hoc applications
 - UPnP applications and services are available over any IP network
 - Ethernet, Wireless LAN, 1394, etc.
- 51 DPnP Architecture Reference
 - · Description/usage
 - Standardized protocols
 - · Standardized XML descriptions
 - · Simple discovery
 - Locate devices/services on-the-fly
 - · Standards-based
- 52 3 How It Works
- 53 System Diagram

54 Wireless Performance

- TCP has many features optimized for wireless in Windows 2000
 - · Improved RTT estimate
 - · Improved window sizes
 - · Fast retransmit
 - Select acknowledgement
 - Acknowledge packets
 - · Improved time-out initiation
 - Very important for wireless losses
 - · Cannot be used over the serial port
 - Use Remote NDIS
 - Over USB, IEEE 1394, Bluetooth wireless technology

55 🔊 WAP

- · WAP was designed to remove some issues with TCP on long thin links
 - · Remove 3 way handshake
 - · Proposals to add data on the SYN and SYN-ACK
 - Reduces DOS protection
- · Remove IP layer for some media
 - · Not removed for GPRS
- · Data compression
 - GPRS supports TCP/IP header and user data compression
 - Recommend GPRS systems support protocol header and user data compression
- · WML is for small screens
 - E.g., a few lines
- 56 Summary Wireless Is Here
 - · Bandwidth is growing
 - · Always connected wireless
 - · Enables new scenarios
 - · Driving new applications
 - · Security a major issue with wireless
 - · 802.1X allows integration into Windows user security system
 - UPnP is the framework for adhoc applications
- 57 Call To Action
 - Mobility
 - · Mediasense is required for roaming support
 - · Any wireless device must generate mediasense
 - Implement 802.1X in network edge devices
 - Switches, access points, etc.
 - · Adhoc services and applications

- · Implement using UPnP
- · Do not limit your applications to a particular wireless media
- 58 Tor More Information
 - · Bluetooth wireless technology
 - http://www.bluetooth.com
 - IrDA
 - http://www.irda.org
 - UPnP
 - http://www.upnp.org
 - http://www.microsoft.com/hwdev/upnp
 - 802.11
 - QoS whitepaper
 - · Security whitepaper
 - · NIC requirements whitepaper
- 59 Tor More Information
 - RNDIS
 - WinHec driver talk
 - http://www.microsoft.com/hwdev/network
 - TCP/IP
 - · Whitepaper
 - http://www.microsoft.com/windows2000/library/howitworks/communications/ networkbasics/tcpip implement.asp
- 60 For More Information
 - IEEE 802.1X
 - http://grouper.ieee.org/groups/802/1/pages/802.1x.html
 - RADIUS
 - http://www.ietf.org/rfc/rfc2138.txt
 - http://www.ietf.org/rfc/rfc2139.txt
 - http://www.ietf.org/rfc/rfc2548.txt
 - http://www.ietf.org/internet-drafts/draft-ietf-radius-radius-v2-06.txt
 - http://www.ietf.org/internet-drafts/draft-ietf-radius-accounting-v2-05.txt
 - http://www.ietf.org/internet-drafts/draft-ietf-radius-ext-07.txt
 - http://www.ietf.org/internet-drafts/draft-ietf-radius-tunnel-auth-09.txt
 - http://www.ietf.org/internet-drafts/draft-ietf-radius-tunnel-acct-05.txt
 - EAP
 - http://www.ietf.org/rfc/rfc2284.txt
 - http://www.ietf.org/rfc/rfc2716.txt

Bluetooth Architecture Overview

Principal Engineer/Bluetooth SIG Chairman Mobile Computer Group Intel Corporation James Kardach



"The Bluetooth Specification is still preliminary; All information regarding Bluetooth is subject to change without notice"

Agenda

- Bluetooth SIG update
- Bluetooth architectural overview
- Summary/call to action

What Does Bluetooth Wireless Technology Do

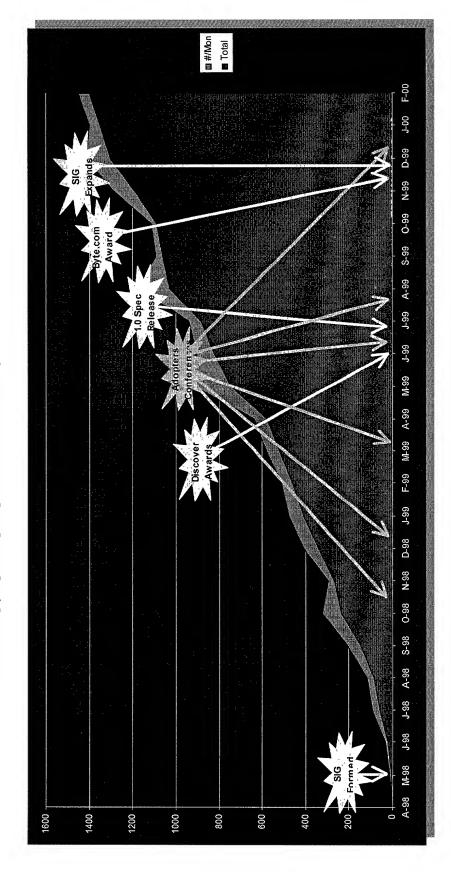
Replacement Cable For You? **Access Points** Data/Voice

Personal Ad-hoc Networks

Program Update

- Final specification published Monday 7/26/99
- Core technology specs and Profile requirements
- Result of work from ~200 engineers
- Bluetooth membership exceeds 1,600 companies!
- Bluetooth wireless technology becoming the choice for wireless connectivity
- Full list of member companies on Web site 25 to 10 logical results
- Bluetooth program on track for products available in 2000
- Products available this year
- Next step is qualification program
- SIG now focusing on ensuring product interoperability
- Bluetooth qualification program started
- Bluetooth wireless technology is the basis for the IEEE 802.15.1 standard
- Bluetooth SIG has expanded
- New contracts and membership types

Bluetooth Momentum



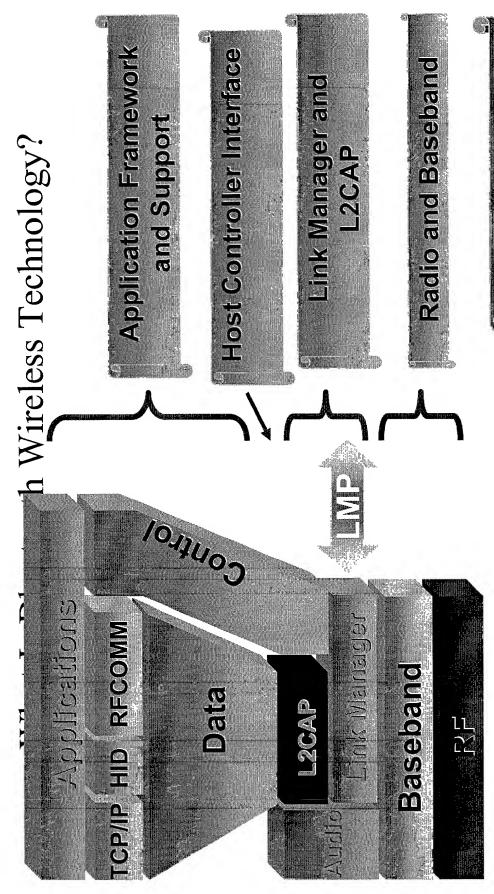
SIG is still growing

The SIG Formally Known As Bluetooth;)

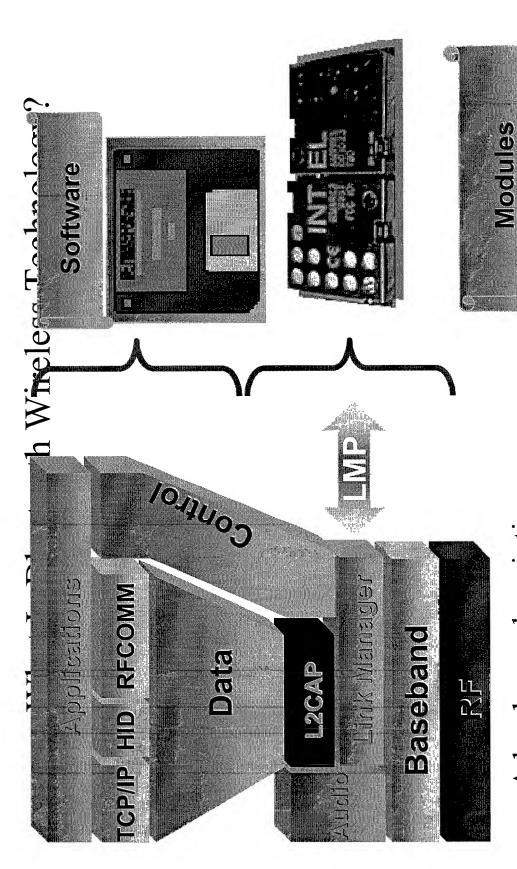
- New Contracts
- Adopter/Early Adopter = Early Adopter
- Early Adopter Contract
- Early Adopter in working group = Associate
- Early Adopter Contract, Associate Amendment
- Open IP license to Bluetooth wireless technology
- Original "Foundation Specifications"
- New technology in and around the 12 specification working groups
- Only need to sign 1 contract to use any Bluetooth wireless technology (the new one)

Future Directions For Bluetooth Profiles

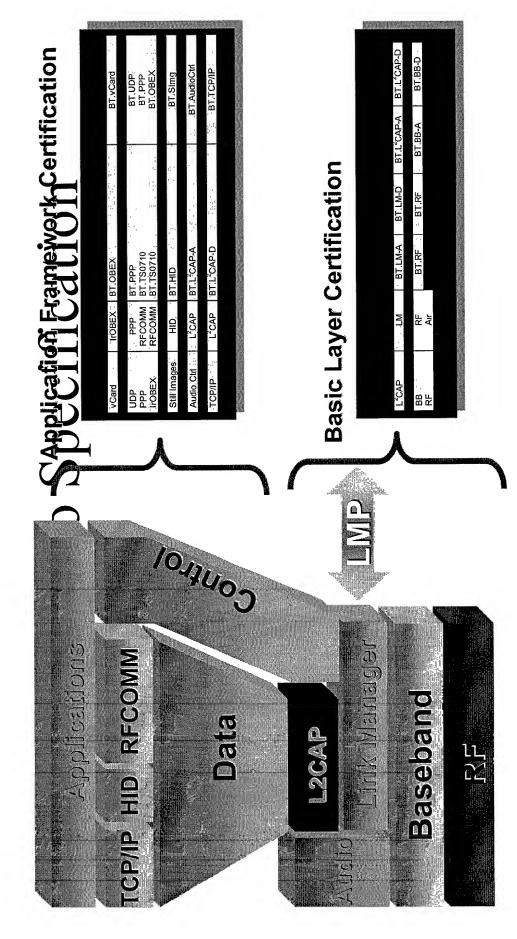
- Second generation radio
- ▶ Personal area networking
- ▶ In and around the car
- . "Wake-up"
- ► Human Interface Devices (HID)
- ◆ Audio/visual
- ► ISM interference/interoperability
- ▶ Printing
- ▶ Still image
- ► Extended Service Discovery protocols
- ◆ Local positioning
- TILL



- A hardware description
 An application framework
- Latest Version on Bluetooth Website: www.Bluetooth.com

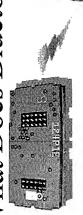


A hardware description
An application framework



- Bluetooth devices will be tested against the specification
- Bluetooth Qualified Test Facilities (BQTF)

What Does Bluetooth Wireless Techno



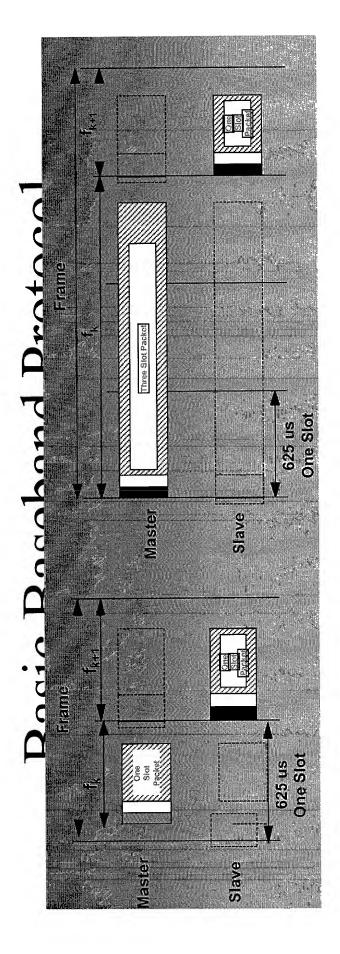


Topology	Supports up to 7 simultaneous links	Each link requires another cable
Flexibility	Goes through walls, bodies, cloths	Line of sight or modified environment
Darka raite	1 제동PS, 720 115.08	Varies प्रक्रिंग पड़न द्यात ट्वंडर
Power	0.1 watts active power	Oyus watts active power or higher
अंटन/भिर्मधुक्षाः	25 mm z Brimns 2 mm, several grams	Size is equal forenge. Typically 1-2 meters, Weightsvaries With length
	ं गांगुन्य स्टाह्म इस्त्री स्टाह्म स्टाह्म	(बाताहुंबड कि छुणातिङ्ग) २ ५५५५पणि/meter (बताचे पडना दण्डर)
्रह्माम्	10 majers of less	Range equal to Size. 1Mpically 1-2 metets
	Intended to work anywhere in the	Calollas vary with local customs
Security	world Very, link layer security, 33 radio	Secure (its a cable)

Cable Replacement

Bluetooth RF Specifications

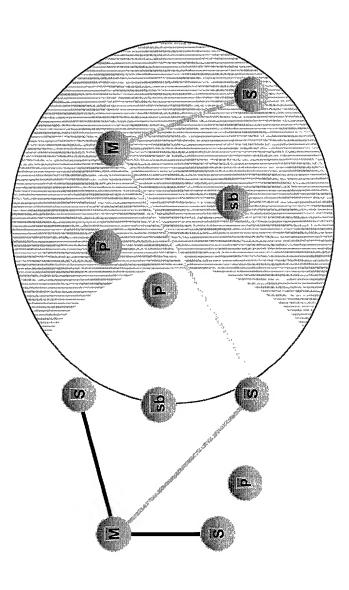
- Specified for low cost, single chip implementation
- Noise floor margin for substrate noise and low current LNA
- Linearity set by near-far problem
- In-band image allows low-cost low IF
- VCO phase noise enables integrated VCO
- TX-RX turn around time enables single synthesizer
- 2.4 ISM band chosen for global use and process capabilities
- Sensitivity traded for low-cost integration of transceiver and Dascopard

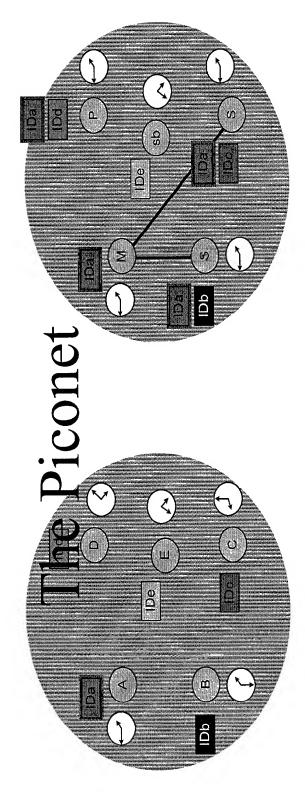


- Spread spectrum frequency hopping radio
- 79/23 one MHz channels
- Hops every packet
- Packets are 1, 3, or 5 slots long
- Frame consists of two packets
- Transmit followed by receive
- Nominally hops at 1600 times a second (1 slot packets)

Network Topology

- Radio Designation
- Connected radios can be master or slave
- Radios are symmetric (same radio can be master or slave)
- Piconet
- Master can connect to 7 simultaneous or 200+ active slaves per piconet
- Each piconet has maximum capacity (1 MSPS)
- Unique hopping pattern/ID
- Scatternet
- High capacity system
- Minimal impact with up to 10 piconets within range
- Radios can share piconets!





- All devices in a piconet hop together
- In forming a piconet, master gives slaves its clock and device ID
 - Hopping pattern determined by device ID (48-bit)
- Phase in hopping pattern determined by Clock
- Non-piconet devices are in standby
- Sonet Addressing
 Active Member Address (AMA, 3-bits) Piconet Addressing
- Parked Member Address (PMA, 8-bits)

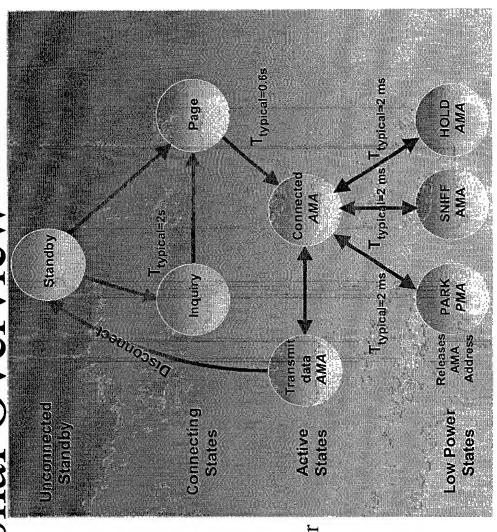


 $\left(\mathsf{qs} \right)$



Functional Overview

- Standby
- Waiting to join a piconet
- Inquire
- Ask about radios to connect to
- Page
- Connect to a specific radio
- Connected
- Actively on a piconet (master or slave)
- Park/Hold/Sniff
- Low Power connected states

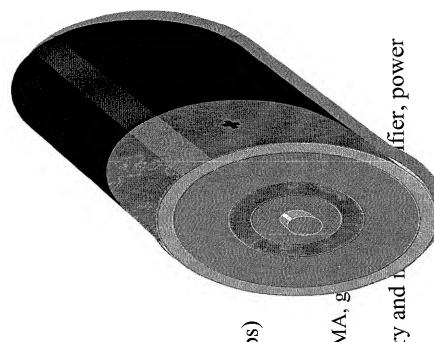


6.601 - 5.601 071 8 7170 9/16 96.7 方形の 36.3 श्रक्रीमामानुद्राह 5/6.0 $= 6.7 M_{\odot}$ 721.00.426 Daita Railes (Kops) Packet Tynes/Data Rates symmetrie 0.1786 432.6 0.301 <u>月</u>1987 256.0 172,3 **E13**(3) 国の人 WOL TON CIVIC DMS EHG. SHI SCO Link Paloket Types 多量 $-/\sqrt{}$ 11/2E) ME 117 0014 01/10 (01/01/ 0000 1.010 0 [-(0(0) 17/06= (0.00) [10]O 30

- ASL –Packet like behavior
- SCO Circuit like behavior

Mobile = Battery Life

- Low power consumption*
- Standby current < 0.3 mA
- b 3 months
- Voice mode 8-30 mA
- b 75 hours
- Data mode average 5 mA
- (0.3-30mA, 20 kbit/s, 25%)
- b 120 hours
- Low Power Architecture
- Programmable data length (else radio sleeps)
- Hold and Park modes 60 µA
- Devices connected but not participating
- Hold retains AMA address, Park releases AMA, g
- Device can participate within 2 ms
- *Estimates calculated with 600 mAh battery and will vary with implementation





Fried Handling 345b

access code

header

payload

Forward-error correction (FEC)

- Headers are protected with 1/3 rate FEC and HEC
- Payloads may be FEC protected
- 1/3 rate: simple bit repetition (SCO packets only)
- 2/3 rate: (10,15) shortened Hamming code
- 3/3 rate: no FEC

ARQ (ACL packets only)

- 16-bit CRC (CRC-CCITT) and 1-bit ACK/NACK
- 1-bit sequence number

Bluetooth Security Features

Fast Frequency Hopping (79 channels)

Low Transmit Power (range <= 10m)

Authentication of remote device

- Based on link key (128 Bit)

- May be performed in both directions

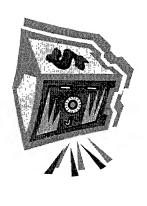
Encryption of payload data

Stream cipher algorithm (≤ 128 Bit)

Affects all traffic on a link

Initialization

- PIN entry by user



Application Level Security

- Builds on-top of link-level security
- Creates trusted device groups
- Security levels for services
- Authorization required
- Authentication required
- Encryption required
- Different or higher security requirements could be added:
- Personal authentication
- Higher security level
- Public key



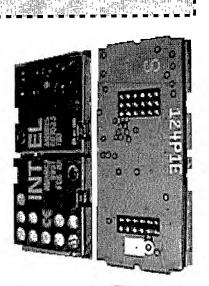
Bluetooth Wireless Technology Is Global

- One version for the world
- Architecture compliant with global emission rules(2.4 GHz ISM band)
- Working through FCC, EC, MPT for spectrum, and power harmonization
- Architecture compliant and safe for use on airlines
- Working with FAA, JAA, FCC, airplane manufacturers, and airlines
- Reviewing security architecture with affected countries



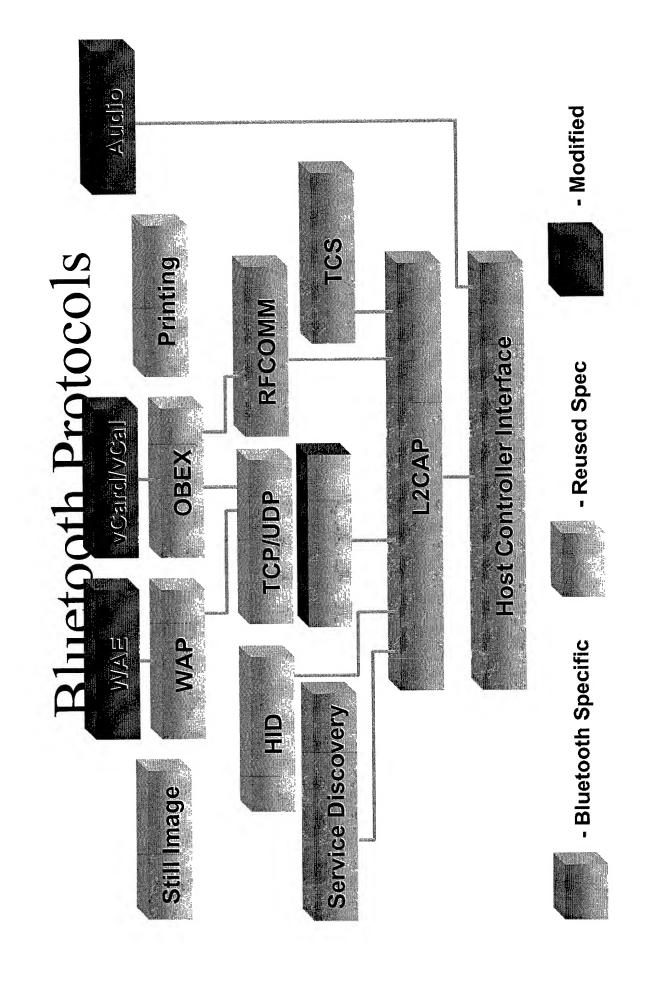
Bluetooth Radio Modules

- Complete radio on a module
- Designed to meet "Limited Module Compliance" (LMA) requirements
- Pre-certified to meet global regulatory requirements
- Allows devices assembled with modules to be "self-certified"
- **USB** Interface
- Solder-ball connections
- External Antennae



Compact

Card



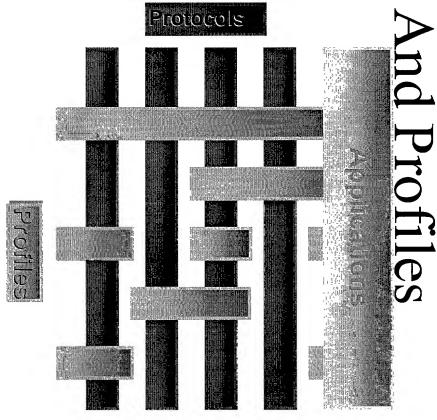
Interoperability And Profiles

 Represents default solution for usage model

 Vertical slice through the protocol stack

Basis for interoperability and logo requirements

Each Bluetooth device supports one or more profiles



Summary

- technology and solution for portable, personal devices Bluetooth wireless technology is a global, RF-based (ISM: 2.4GHz band), short-range, connectivity
- It is not just a radio
- Create piconets on-the-fly (approximately 1Mbps)
- Piconets may overlap in time and space for high aggregate bandwidth
- The Bluetooth spec comprises
- A hardware and software protocol specification
- Usage case scenario profiles and interoperability requirements
- To learn more: http://www.bluetooth.com

Call To Action

- Join the SIG if you haven't already
- Help advance Bluetooth functionality by supporting the working groups committees
- Got a new usage model? Submit a request
- Learn how Bluetooth wireless technology works NOW!
- See Microsoft's presentation on Bluetooth wireless technology
 - Big conference in Monte Carlo check it out!
- More information: The factor of the factor o
- Implement Bluetooth software and hardware in your products and systems
- Insure interoperability via Un-plugfests
- Help support native operating system development
- Provide test hardware to Microsoft

1 🗠	Bluetooth Architecture Overview
	James Kardach Principal Engineer/Bluetooth SIG Chairman Mobile Computer Group Intel Corporation
2 🗀	Agenda
	Bluetooth SIG update
	Bluetooth architectural overview
	Summary/call to action
3 👨	What Does Bluetooth Wireless Technology Do For You?
4 🗀	Program Update
	Final specification published Monday 7/26/99
	Core technology specs and Profile requirements
	 Result of work from ~200 engineers
	Bluetooth membership exceeds 1,600 companies!
	 Bluetooth wireless technology becoming the choice for wireless connectivity
	 Full list of member companies on Web site <u>www.bluetooth.com</u>
	Bluetooth program on track for products available in 2000
	Products available this year
	Next step is qualification program
	 SIG now focusing on ensuring product interoperability
	Bluetooth qualification program started
	 Bluetooth wireless technology is the basis for the IEEE 802.15.1 standard
	Bluetooth SIG has expanded
	New contracts and membership types
5 🔊	Bluetooth Momentum
	SIG is still growing
6 🗀	The SIG Formally Known As Bluetooth ;)
	New Contracts
	 Adopter/Early Adopter = Early Adopter
	Early Adopter Contract
	 Early Adopter in working group = Associate
	 Early Adopter Contract, Associate Amendment
	Open IP license to Bluetooth wireless technology
	 Original "Foundation Specifications"
	 New technology in and around the 12 specification working groups
	 Only need to sign 1 contract to use any Bluetooth wireless technology (the new one)

7 Tuture Directions For Bluetooth ProfilesSecond generation radio

- · Personal area networking
- · In and around the car
- "Wake-up"
- Human Interface Devices (HID)
- Audio/visual
- · ISM interference/interoperability
- Printing
- · Still image
- · Extended Service Discovery protocols
- · Local positioning
- UDI
- - · A hardware description
 - · An application framework
- 9 What Is Bluetooth Wireless Technology?
 - · A hardware description
 - · An application framework
- 10 <a> Testing To Specification
 - · Bluetooth devices will be tested against the specification
 - Bluetooth Qualified Test Facilities (BQTF)
- - · Cable Replacement
- 12 Bluetooth RF Specifications
 - Specified for low cost, single chip implementation
 - Noise floor margin for substrate noise and low current LNA
 - · Linearity set by near-far problem
 - · In-band image allows low-cost low IF
 - VCO phase noise enables integrated VCO
 - TX-RX turn around time enables single synthesizer
 - · 2.4 ISM band chosen for global use and process capabilities
 - · Sensitivity traded for low-cost integration of transceiver and baseband
- 13 <a> Basic Baseband Protocol
 - · Spread spectrum frequency hopping radio
 - 79/23 one MHz channels
 - · Hops every packet
 - · Packets are 1, 3, or 5 slots long
 - · Frame consists of two packets
 - · Transmit followed by receive
 - Nominally hops at 1600 times a second (1 slot packets)
- 14 <a> Network Topology
 - · Radio Designation

- · Connected radios can be master or slave
- · Radios are symmetric (same radio can be master or slave)
- · Piconet
 - Master can connect to 7 simultaneous or 200+ active slaves per piconet
 - Each piconet has maximum capacity (1 MSPS)
 - · Unique hopping pattern/ID
- Scatternet
 - · High capacity system
 - Minimal impact with up to 10 piconets within range
 - Radios can share piconets!
- 15 The Piconet
 - · All devices in a piconet hop together
 - In forming a piconet, master gives slaves its clock and device ID
 - · Hopping pattern determined by device ID (48-bit)
 - Phase in hopping pattern determined by Clock
 - · Non-piconet devices are in standby
 - · Piconet Addressing
 - Active Member Address (AMA, 3-bits)
 - · Parked Member Address (PMA, 8-bits)
- 16 Functional Overview
 - Standby
 - · Waiting to join a piconet
 - Inquire
 - · Ask about radios to connect to
 - Page
 - Connect to a specific radio
 - Connected
 - · Actively on a piconet (master or slave)
 - · Park/Hold/Sniff
 - Low Power connected states
- 17 <a> Packet Types/Data Rates
 - · ASL -Packet like behavior
 - · SCO Circuit like behavior
- 18 Mobile = Battery Life
 - · Low power consumption*
 - Standby current < 0.3 mA
 - Þ 3 months

- Voice mode 8-30 mA
 - Þ 75 hours
- Data mode average 5 mA
- (0.3-30mA, 20 kbit/s, 25%)
 - Þ 120 hours
- · Low Power Architecture
 - Programmable data length (else radio sleeps)
 - Hold and Park modes 60 μA
 - · Devices connected but not participating
 - · Hold retains AMA address, Park releases AMA, gets PMA address
 - · Device can participate within 2 ms
 - *Estimates calculated with 600 mAh battery and internal amplifier, power will vary with implementation
- 19 🗷 Error Handling
 - Forward-error correction (FEC)
 - Headers are protected with 1/3 rate FEC and HEC
 - · Payloads may be FEC protected
 - 1/3 rate: simple bit repetition (SCO packets only)
 - 2/3 rate: (10,15) shortened Hamming code
 - 3/3 rate: no FEC
 - ARQ (ACL packets only)
 - 16-bit CRC (CRC-CCITT) and 1-bit ACK/NACK
 - 1-bit sequence number
- 20 <a> Bluetooth Security Features
 - Fast Frequency Hopping (79 channels)
 - Low Transmit Power (range <= 10m)
 - · Authentication of remote device
 - Based on link key (128 Bit)
 - · May be performed in both directions
 - · Encryption of payload data
 - Stream cipher algorithm (≤ 128 Bit)
 - · Affects all traffic on a link
 - Initialization
 - · PIN entry by user
- 21 <a> Application Level Security
 - · Builds on-top of link-level security
 - · Creates trusted device groups
 - · Security levels for services
 - · Authorization required
 - · Authentication required
 - · Encryption required

- Different or higher security requirements could be added:
 - · Personal authentication
 - · Higher security level
 - Public key
- 22 3 Bluetooth Wireless Technology Is Global
 - · One version for the world
 - Architecture compliant with global emission rules (2.4 GHz ISM band)
 - Working through FCC, EC, MPT for spectrum, and power harmonization
 - · Architecture compliant and safe for use on airlines
 - · Working with FAA, JAA, FCC, airplane manufacturers, and airlines
 - · Reviewing security architecture with affected countries
- 23 <a> Bluetooth Radio Modules
 - · Complete radio on a module
 - Designed to meet "Limited Module Compliance" (LMA) requirements
 - · Pre-certified to meet global regulatory requirements
 - · Allows devices assembled with modules to be "self-certified"
 - · USB Interface
 - · Solder-ball connections
 - External Antennae
- 24 🔊 Bluetooth Protocols
- 25 <a> Interoperability And Profiles
 - · Represents default solution for usage model
 - Vertical slice through the protocol stack
 - Basis for interoperability and logo requirements
 - Each Bluetooth device supports one or more profiles
- 26 Summary
 - Bluetooth wireless technology is a global, RF-based (ISM: 2.4GHz band), short-range, connectivity technology and solution for portable, personal devices
 - · It is not just a radio
 - Create piconets on-the-fly (approximately 1Mbps)
 - Piconets may overlap in time and space for high aggregate bandwidth
 - · The Bluetooth spec comprises
 - · A hardware and software protocol specification
 - · Usage case scenario profiles and

interoperability requirements

• To learn more: http://www.bluetooth.com

27 Call To Action

- Join the SIG if you haven't already
 - Help advance Bluetooth functionality by supporting the working groups committees
 - · Got a new usage model? Submit a request
- · Learn how Bluetooth wireless technology works NOW!
 - · See Microsoft's presentation on Bluetooth wireless technology
 - Big conference in Monte Carlo check it out!
 - More information: http://www.Bluetooth.com
- · Implement Bluetooth software and hardware in your products and systems
 - · Insure interoperability via Un-plugfests
- Help support native operating system development
 - · Provide test hardware to Microsoft

Bluetooth User Experience in Windows

Dr. Michael W. Foley Software Design Engineer Microsoft Corporation

High Level Goals

- Easy configuration
- Configure a new device once
- Wizard to guide user through the process
- Ease of use
- After initial configuration, it "just works"
- Known devices can interoperate without user intervention
- Don't overwhelm user with unnecessary information
- Enable advanced functionality
- User configurable security
- Local radio properties

User Scenarios Enabled

- Cell phone as a modem
- Dial-up networking profile
- Digital camera to PC
- File transfer profile
- Public Internet access points
- LAN Access Profile
- Business card exchange
- Object push profile
- Synchronize PPC with desktop

Configuration Issues

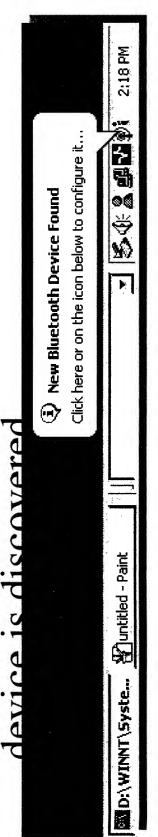
- Security modes
- Open device no security
- Secure connections only
- Service level security
- Binding devices
- PIN entry on multiple devices
- Support for devices without a UI
- Enable hidden computing
- How to hide this complexity from the average user?

Device States

- Unknown device in range
- Known, unbound device in range
- Bound device in range
- Device out of range

Unknown Device In Range

- Device discovery
- Periodic polling can be enabled
- User can force a discovery
- Unobtrusive notification will appear when a



Managing Devices

- Approve device
- Establish trust relationship
- Requires entering data on each device
- One-time procedure
- Establish temporary connection
- One-time service usage
- May or may not be secure connection
- Instruct Windows to ignore this device in the future
- Stop telling me about cell phone in the office next to mine

Approval Wizard

- Guides the user through the binding process
- Welcome page
- Brief explanation of what the user needs to
- Specifically calls out that devices must be explicitly enabled

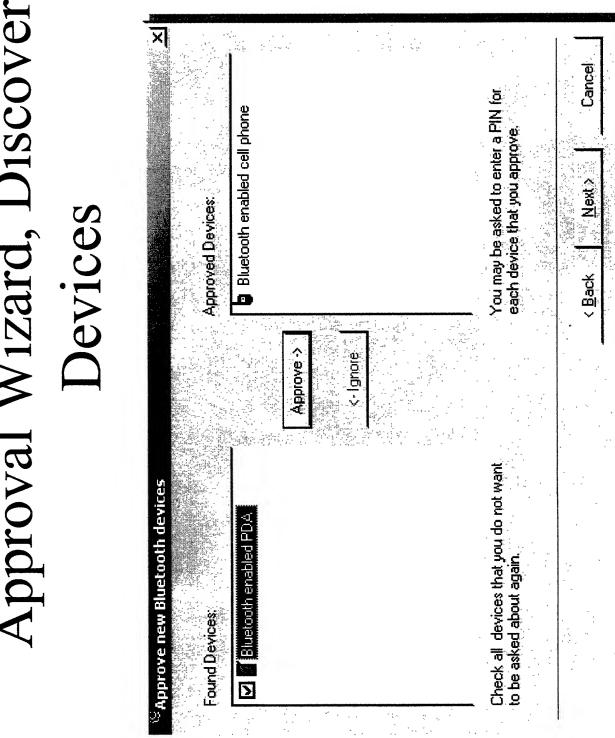
Approval Wizard, Welcome

ove New Bluetooth Devices

Welcome to the Bluetooth device approval wizard.

You must explicitly choose to enable newly discovered devices; by default, all devices are not approved.

Approval Wizard, Discovered



Approve New Bluetooth Devices Wilzard, Key Ent Enter the PIN for each approved Bluetooth device: Bluetooth enabled cellphone

Approval Wizard, Remote Device Instructions

You have finished approving the newly discovered Bluetooth devices.

You must now enter the PIN for this computer on each of the discovered device

< Back Finish

Cannel

Approved Device In Range

- User can immediately use services exported by the device
- PIN not required
- Supports "Hidden Computing" model
- PPC automatically syncs with desktop when in range

Dialup Networking

Use the Network Connection Wizard to create a connectiod Select the Bluetooth Device that appears as a modem

If a connectiod already exist, simply change to the Bluetooth modem device

Connect to network just like any other modem

V Show icon in taskbar when connected
Lise dialing rules
Country/region code:
Area code:
V All devices call the same number
Modem - Worldport PCMCIA 144UU (USH command
Connect using:
Local 1SP General Options Security Networking

LAN Access Point

- connection LAN Access Point looks like a modem
- Usage is the same as DUN, except the point rather than the phone/modem connectoid connects to the LAN access

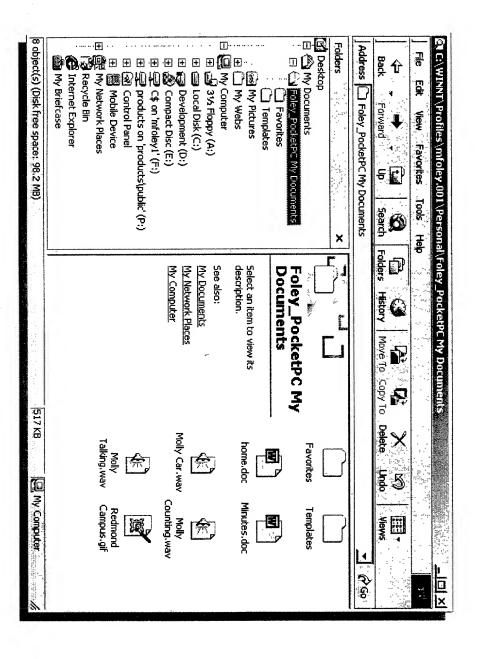
File Transfer

Browser remote device

10 object(s)	Temporary Internet Files	⊞. ☐ My Documents ⊞. ☐ Program Files ☐ Temp	products on 'products\public' (P:) Control Panel	Folders × C\$ on 'mfoley1' (F:)	Address Mobile Device	← + € Back Forward Up Search F	File Edit View Favorites Tools Help	@\Foley_PocketPC
		Bridge.jpg	My Documents Program Files			Folders History		
Foley_PocketPC		cehello.exe	Program Files			Move To Copy To	The state of the s	
cketPC	4	Home.jpg	Temp			Pro Dejete		
		Restart Synchroniza	Temporary Internet Files			5 5		11.5 2.7 2.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3.7 3
		art San niza Francisco.htm	Windows		₹	Views		

Synchronization

File as well as contacts, appointments, task, etc.



Advanced Functionality

- Listing all devices in range
- Managing local services
- Managing local radio parameters
- Removing devices
- Forced discovery

- Remote Device Browsing
 User can view all approved devices; both in and out of range
- Can remove device from approved status
- currently in range User can only view unapproved devices
- History of all devices seen isn't stored
- Explore Services on remote devices
- Property page for each device will display for the device URLs associated with service discovery record

Managing Local Services

- View currently exported services
- Alter security settings
- Must be bound device by default
- Enable access to unbound devices
- Disable service to all devices

Managing Local Radio Properties

- Set power management policies
- AC Powered versus Battery Powered
- Periodic polling frequency
- Machine is on
- Machine is in standby
- Machine is in hibernate
- Which COD wake machine
- Set filtering rules
- Discover particular CODs
- Discover particular radios
- Some properties are read-only for non administrators

Summary

- Designed for ease of use
- Configure devices once
- More information and control available when required
- Integrated Windows User Interface
- Many usage models supported out of the box
- Dial-up networking
- Synchronization
- File Transfer
- LAN Access

Call To Action

- Beta test our software
- Learn first hand what is intuitive, and what is
- Which features should be more readily available
- Which features are rarely used
- Provide us feedback
- Provide us hardware
- Allows us to test the user experience with a wide range of scenarios

1 🗂	Bluetooth User Experience in Windows				
2 🗖	Dr. Michael W. Foley Software Design Engineer Microsoft Corporation High Level Goals Easy configuration Online a new device once				
	Wizard to guide user through the process				
	• Ease of use				
	After initial configuration, it "just works"				
	Known devices can interoperate without user intervention				
	Don't overwhelm user with unnecessary information				
	Enable advanced functionality				
	User configurable security				
	Local radio properties				
3 🗀	User Scenarios Enabled				
	Cell phone as a modem				
	Dial-up networking profile				
	Digital camera to PC				
	File transfer profile				
	Public Internet access points				
	LAN Access Profile				
	Business card exchange				
	Object push profile				
_	Synchronize PPC with desktop				
4 🗀	•				
	Security modes				
	Open device – no security				
	Secure connections only				
	Service level security				
	Binding devices				
	PIN entry on multiple devices				
	Support for devices without a UI - No hidden computing				
	Enable hidden computing How to hide this complexity from the gyerage user?				
5 🗂	How to hide this complexity from the average user? Povice States				
ل د	Device States				
	Unknown device in rangeKnown, unbound device in range				
	Bound device in range				

Device out of range

6 D Unknown Device In Range Device discovery · Periodic polling can be enabled · User can force a discovery · Unobtrusive notification will appear when a device is discovered 7 Managing Devices · Approve device · Establish trust relationship Requires entering data on each device · One-time procedure · Establish temporary connection · One-time service usage · May or may not be secure connection · Instruct Windows to ignore this device in the future · Stop telling me about cell phone in the office next to mine 8 Approval Wizard · Guides the user through the binding process Welcome page · Brief explanation of what the user needs to do · Specifically calls out that devices must be explicitly enabled 9 Approval Wizard, Welcome 10 <a> Approval Wizard, Discovered Devices 11 <a> Approval Wizard, Key Entry 12 <a> Approval Wizard, Remote Device Instructions 13 Approved Device In Range · User can immediately use services exported by the device PIN not required · Supports "Hidden Computing" model · PPC automatically syncs with desktop when in range 14 Dialup Networking 15 LAN Access Point LAN Access Point looks like a modem connection · Usage is the same as DUN, except the connectoid connects to the LAN access point rather than the phone/modem 16 🔊 File Transfer · Browser remote device · Drag and drop files 17 Synchronization · File as well as contacts, appointments, task, etc.

18 Advanced Functionality

· Listing all devices in range

- Managing local services
- · Managing local radio parameters
- · Removing devices
- · Forced discovery
- 19 Remote Device Browsing
 - · User can view all approved devices; both in and out of range
 - · Can remove device from approved status
 - · User can only view unapproved devices currently in range
 - · History of all devices seen isn't stored
 - Explore Services on remote devices
 - Property page for each device will display URLs associated with service discovery record for the device
- 20 Managing Local Services
 - · View currently exported services
 - · Alter security settings
 - · Must be bound device by default
 - · Enable access to unbound devices
 - · Disable service to all devices
- 21 Managing Local Radio Properties
 - · Set power management policies
 - · AC Powered versus Battery Powered
 - · Periodic polling frequency
 - · Machine is on
 - · Machine is in standby
 - · Machine is in hibernate
 - · Which COD wake machine
 - · Set filtering rules
 - Discover particular CODs
 - · Discover particular radios
 - · Some properties are read-only for non administrators
- 22 Summary
 - · Designed for ease of use
 - Configure devices once
 - More information and control available when required
 - Integrated Windows User Interface
 - Many usage models supported out of the box
 - · Dial-up networking
 - Synchronization
 - · File Transfer
 - LAN Access
- 23 Call To Action

- · Beta test our software
 - Learn first hand what is intuitive, and what is not
 - Which features should be more readily available
 - Which features are rarely used
- Provide us feedback
- · Provide us hardware
 - · Allows us to test the user experience with a wide range of scenarios

Elusiooih Siack In Windows

Software Development Engineers प्रधा प्रधार आत्र डोडात प्रबेश प्रधान Microsoft Corporation Windows Division

Agenda

- Bluetooth Architecture in Windows
- Goals
- Components of the Stack
- Functionality
- Opportunities for IHVs and ISVs
- Applications
- Services
- Devices

High Level Goals

- PC work with all devices
- Bluetooth Devices as PC peripherals
- Bluetooth Devices as PC companions
- Bluetooth Devices bridge to network resources through a PC
- Easy to configure and operate
- Extensible architecture
- Platform for third parties to add value

Scenarios

- Device configuration:
- Discovery
- Bonding
- Syncing and transfer through OBEX
 - Files
- Pictures
- Vcards
- Dial up Networking
 - Cell as modem
- Null Modem for Peer to peer
- Generic RFComm applications
 - Non-OBEX synchronization
- Other serial-type applications

Technical Requirements

- Bluetooth 1.0 Type II device classification supported
- Required profiles
- Bus Management Infrastructure
- Device and radio configuration
- Control panels
- System Trays
- Extensible framework for value adds
- Devices
- Profiles
- Bus mgmt software
- RFComm applications
- Object Exchange and special object handling
- RAS and TAPI over Unimodem

Bluetooth Stack Diagram

OBEX.DLL

WITSOCKZ

WODEMISKS

できる かい

RESONN'SYS

CICION

SDP/Advisor

Stack Components

- BthPort
- L2Cap / HCI
- Hardware abstraction: Serial, USB...
- Enumeration of Found Bound Services
- SDP/Management UI
- Bus management:
- User notification of newly discovered devices
- User assisted Configuration and Bonding
- Configuration of radio
- Local Service Exposure and Publication

Stack Components

- RFCOMM
- RFComm Profile
- TDI interface for WinSock (AFD)
- Bus enumeration for DUNs
- BthModem (a WDM modem)
- OBEX.DLL
- Object Exchange 1.2
- Bus Agnostic

BthPort

Support Currently Defined buses: USB, Serial, 16550

• Plug and Play events

• Bluetooth Request Blocks

SDP

Provide a "builder" interface to easily create a service record

Kernel mode

- newly discovered devices or initiate a SDP search outside of Client drivers can submit a list of UUIDs to search for on all device discovery
- BThPort will search for all the services in the browse group hierarchy

• User mode

- Initiate searches
- Browse service records

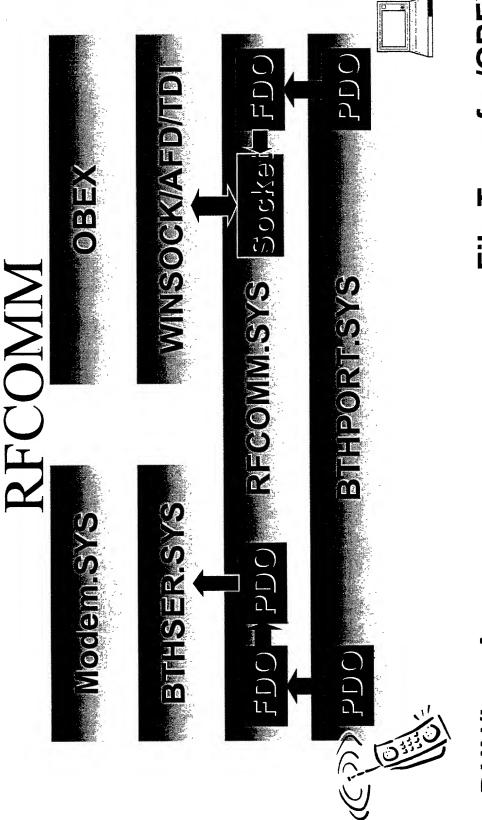
Management UI

- Present user with devices in range and bound devices
- Allows the user to easily change the relationship with remote device
- Provide unobtrusive PIN and authorization notifications
- UI is accessible from third-party applications for a standard user experience
- Advanced features
- Filter devices based on COD or address
- Local radio settings
- Manage power policies

OBEX

- Full OBEX 1.2 implementation:
- Put
- Get
- SetPath
- Definable transactions
- COM API
- Extensible to other media and transports

WINSOCK 2 OBEX OBEXIFD/A.DLL



DUN/Lan Access

File Transfer/OBEX

Opportunities To Add Value

- RF comm applications
- OBEX applications/extensions
- Bluetooth management application
- New device types and/or class drivers
- Radios on new hardware buses

RF Comm Applications

- Applications looking for virtual serial ports not supported
- Legacy TAPI/Unimodem applications see peer devices as NULL Modems
- Applications enumerate Modem/Serial Devices through Unimodem

RF Comm Applications

- Winsock allows for dynamic discovery and communication
- Talk to the device, not to the conduit ("My Laserjet" versus LPT2)
- Once bonded device is in range the application can find and use it
- Allows for multiple remote connection to same service.
- Not necessary to manage multiple virtual COMx ports

OBEX Applications

- Examples
- Photos
- Vcards (not "in the box")
- Simple databases
- Server
- Registration
- New Obex Commands and types
- Application can register as handler for custom commands
- Client
- Discovery
- Navigate directory structure (enumerate objects)
- Push Pull objects

Bluetooth Management Applications

- Substitution of stock Microsoft Plug and Play experience
- Configuration and bonding of devices

New Profiles Types

- Native L2CAP
- Examples:
- HID
- Remote NDIS
- Doom Server with Streaming Audio (utilize native audio channels)

New Profile Types

Server

- Registers with SDP/Adviser module
- No Plug and Play event until remote peer connects
- PDO means active connection on which local server driver loaded

Client

- BthPort finds remote service
- wants to use this device (Approval Wizard) Signals SDP/Advisor to determine if user
- Plug and Play event (PDO) means active connection on which local client driver loaded

New Hardware Buses

Examples

Register set for Card Bus/PCI

-1394

Miniport

Calls To Action

- Is your value add method missing?
- We need your hardware
- Phones
- Radios
- Phones
- Peripherals
- Phones
- PDAs
- We need your software
- Applications and drivers
- Can we upgrade you?
- Come to the developers' conference

References

- http://www.microsoft.com/ hwdev/bluetooth
- Contact: BTINFO@Microsoft.COM

THE REPORT OF THE PERSON HAVE BEEN AND THE PERSON OF THE PERSON HAVE BEEN AND THE PERSON HAVE BE

Windows Hardware Engineering Conference

licrosoft INST

. (=)	
1 🔊	Arranda
2 🗀	Agenda • Bluetooth Architecture in Windows
	Goals Components of the Stock
	Components of the Stack Transferred the Stack
	Functionality Opportunities for IHVs and ISVs
	Opportunities for IHVs and ISVs Applications
	ApplicationsServices
	Devices
3 🗐	High Level Goals
ت د	PC work with all devices
	Bluetooth Devices as PC peripherals
	Bluetooth Devices as PC companions
	Bluetooth Devices bridge to network resources through a PC
	Easy to configure and operate
	Extensible architecture
	Platform for third parties to add value
4 🗀	
	Device configuration:
	• Discovery
	• Bonding
	Syncing and transfer through OBEX
	• Files
	Pictures
	Vcards
	Dial up Networking
	Cell as modem
	Null Modem for Peer to peer
	Generic RFComm applications
	 Non-OBEX synchronization
	Other serial-type applications
5 🗀	Technical Requirements
	 Bluetooth 1.0 Type II device classification supported
	Required profiles
	Bus Management Infrastructure
	 Device and radio configuration
	Control panels
	System Trays

• Extensible framework for value adds

• Devices

- Profiles
- · Bus mgmt software
- · RFComm applications
- · Object Exchange and special object handling
- · RAS and TAPI over Unimodem
- 6 Bluetooth Stack Diagram
- 7 Stack Components
 - BthPort
 - · L2Cap / HCl
 - · Hardware abstraction: Serial, USB...
 - Enumeration of Found Bound Services
 - SDP/Management UI
 - · Bus management:
 - · User notification of newly discovered devices
 - · User assisted Configuration and Bonding
 - Configuration of radio
 - · Local Service Exposure and Publication
- 8 Stack Components
 - RFCOMM
 - RFComm Profile
 - TDI interface for WinSock (AFD)
 - · Bus enumeration for DUNs
 - BthModem (a WDM modem)
 - · OBEX.DLL
 - Object Exchange 1.2
 - Bus Agnostic
- 9 🗀 BthPort
 - Support Currently Defined buses: USB, Serial, 16550
 - · Plug and Play events
 - Bluetooth Request Blocks
- 10 SDP
 - Provide a "builder" interface to easily create a service record
 - Kernel mode
 - Client drivers can submit a list of UUIDs to search for on all newly discovered devices or initiate a SDP search outside of device discovery
 - · BThPort will search for all the services in the browse group hierarchy
 - User mode
 - · Initiate searches
 - · Browse service records
- 11 Management UI
 - Present user with devices in range and bound devices

- · Allows the user to easily change the relationship with remote device
- Provide unobtrusive PIN and authorization notifications
- · UI is accessible from third-party applications for a standard user experience
- · Advanced features
 - · Filter devices based on COD or address
 - Local radio settings
 - · Manage power policies
- 12 OBEX
 - Full OBEX 1.2 implementation:
 - Put
 - Get
 - SetPath
 - · Definable transactions
 - COM API
 - Extensible to other media and transports
- 13 🔊 OBEX
- 14 🔊 RFCOMM
- 15 Opportunities To Add Value
 - · RF comm applications
 - OBEX applications/extensions
 - Bluetooth management application
 - · New device types and/or class drivers
 - · Radios on new hardware buses
- 16 RF Comm Applications
 - · Applications looking for virtual serial ports not supported
 - · Legacy TAPI/Unimodem applications see peer devices as NULL Modems
 - Applications enumerate Modem/Serial Devices through Unimodem
- 17 RF Comm Applications
 - · Winsock allows for dynamic discovery and communication
 - Talk to the device, not to the conduit ("My Laserjet" versus LPT2)
 - · Once bonded device is in range the application can find and use it
 - · Allows for multiple remote connection to same service
 - Not necessary to manage multiple virtual COMx ports
- 18 OBEX Applications
 - Examples
 - Photos
 - Vcards (not "in the box")
 - · Simple databases
 - Server
 - Registration

- New Obex Commands and types
- · Application can register as handler for custom commands
- Client
 - Discovery
 - · Navigate directory structure (enumerate objects)
 - Push Pull objects
- 19 🗂 Bluetooth Management Applications
 - Substitution of stock Microsoft Plug and Play experience
 - · Configuration and bonding of devices
- 20 New Profiles Types
 - Native L2CAP
 - · Examples:
 - · HID
 - Remote NDIS
 - Doom Server with Streaming Audio (utilize native audio channels)
- 21 New Profile Types
 - Server
 - · Registers with SDP/Adviser module
 - No Plug and Play event until remote peer connects
 - PDO means active connection on which local server driver loaded
 - Client
 - · BthPort finds remote service
 - Signals SDP/Advisor to determine if user wants to use this device (Approval Wizard)
 - Plug and Play event (PDO) means active connection on which local client driver loaded
- 22 New Hardware Buses
 - Examples
 - · Register set for Card Bus/PCI
 - 1394
 - Miniport
- 23 Calls To Action
 - · Is your value add method missing?
 - · We need your hardware
 - Phones
 - Radios
 - · Phones
 - Peripherals
 - Phones
 - PDAs

- We need your software
 - Applications and drivers
 - Can we upgrade you?
- Come to the developers' conference
- 24 TReferences
 - http://www.microsoft.com/ hwdev/bluetooth
 - Contact: BTINFO@Microsoft.COM

25 🗂